KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Vytautas Mickevičius

Elektroninės komercijos programinė įranga mobiliesiems įrenginiams: analizė bei kūrimas

Magistro darbas

Vadovas
doc. dr. E. Bareiša

KAUNAS, 2005

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas
doc. dr. E. Bareiša

Elektroninės komercijos programinė įranga mobiliesiems įrenginiams: analizė bei kūrimas

Informatikos mokslo magistro baigiamasis darbas

Kalbos konsultantė
Lietuviu kalbos katedros lektorė
dr. J. Mikelionienė

Vadovas
doc. dr. E. Bareiša

Recenzentas
doc. dr. E. Karčiauskas

Atliko
IFM 9/7 gr. stud.
V.Mickevičius

KAUNAS, 2005

TURINYS

1 ĮVADAS 9

1.1 Tyrimo sritis, objektas ir problema 9

2 ELEKTRONINĖS KOMERCIJOS IR WAP ANALIZĖ 11

2.1 Elektroninės komercijos analizė 12

2.1.1 Elektroninės komercijos samprata 12

2.1.2 Elektroninės komercijos formos, teikiama nauda 13

2.1.3 Tipinis elektroninės komercijos atvejis - elektroninė parduotuvė 14

2.2 WAP protokolo galimybių analizė 15

2.2.1 WAP technologijos apžvalga 17

2.2.2 WAP architektūros apžvalga 18

2.2.3 WAP saugumas - WTLS 24

2.2.4 LEAP – alternatyva WAP 25

2.3 Analizės bei projektavimo priemonių parinkimas 26

2.4 Programinės įrangos, technologijų bei architektūros pasirinkimas 27

2.5 Analizės išvados 28

3 ELEKTRONINĖS KOMERCIJOS SPRENDIMŲ TAIKYMO WAP TECHNOLOGIJOSE METODIKA 30

3.1 Sprendimų taikymo WAP pagrindimas 31

3.2 Sprendimų taikymo WAP kokybės kriterijai 32

3.3 Sprendimų taikymo WAP specifika 33

3.4 Infrastruktūros pakeitimai reikalingi WAP taikymui 34

3.5 Sprendimų taikymo WAP architektūra 35

3.5.1 Tipinė architektūra 36

3.5.2 Architektūra su išskirtu atvaizdavimo sluoksniu 36

3.5.3 Architektūra su kodavimo serveriu 37

3.5.4 Architektų sluoksninė analizė 38

3.6 Sprendimų taikymo WAP atvejai 39

3.6.1 Esami sprendimai 40

3.6.2 Nauji sprendimai 40

3.7 HTML vertimas į WML 41

3.7.1 Vertimo metodika 42

3.7.2 Žymų vertimas 43

3.7.3 Grafinių elementų vertimas WAP aplinkai 48

3.7.4 Vartotojo sąsajos elementų vertimas 49

3.8 Kitos sprendimų taikymo WAP problemos 50

3.9 Sprendimų taikymo WAP išvadų santrumpa ... 51
3.10 Vartotojo agento nustatymas ... 52
3.11 Prototipo kūrimas – WWW terpė ... 52
3.11.1 Organizacijos modelis ... 54
3.11.2 Esybių sąryšių diagrama ... 55
3.11.3 Objektiniai modeliai ... 55
3.11.4 Kanoninė schema ... 59
3.11.5 Objektų-savybių modelis ... 60
3.11.6 Duomenų bazės struktūra ... 60
3.12 Prototipo kūrimas – WAP terpė ... 62
3.12.1 Naujos terpės įtraukimas į sprendimą ... 62
3.12.2 Žymų vertimas ... 63
3.12.3 Grafinių elementų vertimas ... 64
4 ELEKTRONINĖS KOMERCIJOS SPRENDIMO PROTOTIPO PRITAIKYMO WAP TERPEI REZULTATŲ TYRIMAS ... 65
4.1 Principinė galimybė pritaikyti elektroninės komercijos sprendimą WAP ... 65
4.2 Funkciniai pokyčiai elektroninės komercijos sprendimą pritaikius WAP ... 65
4.3 Vartotojo sąsajos pokyčiai elektroninės komercijos sprendimą pritaikius WAP ... 68
5 IŠVADOS ... 72
6 LITERATŪRA ... 73
7 TERMINŲ IR SANTRUMPŲ ŽODYNAS ... 75
8 PRIEDAI ... 76
8.1 WAP protokolo galimybių analizė ... 76
8.1.1 Įžanga į WAP ... 76
8.1.2 WAP technologijos apžvalga ... 77
8.1.3 WAP Forum siekiami architektūros tikslai ... 79
8.1.4 WAP architektūros apžvalga ... 80
8.1.5 WAP architektūros komponentai ... 87
8.1.6 Belaidžių datagamų protokolo WDP architektūros apžvalga ... 92
8.1.7 Belaidžio transporto sluoksnio saugumas WTLS ... 109
8.1.8 Belaidžių sesijų protokolas WSP ... 128
8.1.9 Pavyzdinės WAP technologijos konfigūracijos ... 131
8.1.10 LEAP – alternatyva WAP ... 134
8.1.11 Išvados ... 139
8.2 Elektroninės komercijos analizė ... 140
8.2.1 Elektroninės komercijos samprata ... 140
8.2.2 Elektroninės komercijos kategorijos ... 142
8.2.3 Elektroninės komercijos įtaka ... 143

8.2.4 Elektroninės komercijos veikla 143
8.2.5 Tiekėjų galimybės ir klientų nauda 145
8.2.6 Elektroninės komercijos pavyzdžiai Lietuvoje 146
8.2.7 Elektroninės komercijos sprendimai 149
8.3 Elektroninės parduotuvės veiklos analizė 150
8.3.1 Įvadas 150
8.3.2 Elektroninės parduotuvės panaudojimo atvejų modelis 150
8.3.3 Veiklos analizės išvados 152

LENTELIŲ SĄRAŠAS

2.1 lentelė WAP ir LEAP palyginimas ... 26
3.1 lentelė Minimalių WML ir HTML dokumentų pavyzdžiai ... 34
3.2 lentelė WAP naudojami MIME tipai ... 35
3.3 lentelė Sąsajų tarpusavio komunikavimo protokolai ... 39
3.4 lentelė HTML žymų vertimo į WML žymas taisyklių lentelė ... 44
3.5 lentelė WML elementų aukštesniųjų elementų lentelė. ... 48
3.6 lentelė Tipinių HTML sąsajos elementų vertimas į WML ... 50
3.7 lentelė Supaprastintas atvaizdavimo sluoksnio metodo vertimas iš HTML į WML ... 63
3.8 lentelė Atvaizdavimo sluoksnio metodų sugeneruotų rezultatų pavyzdys ... 64
4.1 lentelė Funkcionalumo pokyčiai WAP taikyme ... 67
4.2 lentelė Vartotojo sąsajos pokyčiai WAP taikyme ... 69

PAVEIKSLŲ SĄRAŠAS

2.1 pav. Elektroninės komercijos skirstymas pagal bendraujančias šalis ... 13
2.2 pav. Elektroninės komercijos skirstymas pagal taikymo sferą ... 13
2.3 pav. Elektroninės komercijos teikiama nauda ... 14
2.4 pav. Bendrasis panaudojimo atvejų modelis ... 14
2.5 pav. Prekių sąrašo modifikavimo modelis ... 15
2.6 pav. Vartotojų sąrašo modifikavimo modelis ... 15
2.7 pav. Pasaulinio žiniatinklio programavimo modelis ... 19
2.8 pav. WAP programavimo modelis ... 20
2.9 pav. Savybes bei našumą pagerinantys tarpiniai serveris ... 21
2.10 pav. Pagalbiniai serveriai ... 21
2.11 pav. Pavyzdinė WAP tinklo schema ... 22
2.12 pav. WAP kliento architektūra ... 23
2.13 pav. WAP dėklo architektūra ... 24
3.1 pav. Pagrindinės WAP ir WWW žiniatinklio technologijos ... 30
3.2 pav. Įprastinė WAP panaudojimo schema, kurioje matoma tarpinio serverio pozicija ... 33
3.3 pav. Egzistuojančios architektūros panaudojimas WAP prieigai ... 36
3.4 pav. Architektūros su išskirtu atvaizdavimo sluoksniu panaudojimas WAP prieigai ... 36
3.5 pav. Architektūros su kodavimo serveriu panaudojimas WAP prieigai ... 37
3.6 pav. Apibendrinta 4 sluoksnių sprendimo schema ... 39
3.7 pav. Paprasto grafinio elemento vertimas į WBMP ... 49
3.8 pav. Vertimas į WBMP – paprastas bei naudojant sklaidą (angl. *error diffusion*) ... 49
3.9 pav. Organizacijos modelis ... 54
3.10 pav. Esybių sąryšių diagrama ... 55
3.11 pav. Klasių diagrama ... 55
3.12 pav. Panaudojimo atvejų diagrama ... 56
3.13 pav. Bendradarbiavimo diagramos – prekes pasirinkimo diagrama ... 56
3.14 pav. Bendradarbiavimo diagramos – lankytojo registracijos diagrama ... 56
3.15 pav. Bendradarbiavimo diagramos – užsakymo diagrama ... 56
3.16 pav. Prekes pasirinkimo diagrama ... 57
3.17 pav. Lankytojo registracijos diagrama ... 57
3.18 pav. Užsakymo diagrama ... 57
3.19 pav. Būsenų diagrama ... 58
3.20 pav. Kanoninė schema ... 59
3.21 pav. Objektų-savybių modelis ... 60
3.22 pav. Automatinio grafinių elementų vertimo, rezultato pavyzdys ... 64

4.1 pav. Organizacijos modelis WAP atvejui 66
4.2 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – pradinis puslapis 68
4.3 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – paieška prekių sąraše 69
4.4 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – prekių kategorijų peržiūra..... 70
4.5 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – prekės peržiūra..... 70
4.6 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – prekės užsakymas 71

E-COMMERCE SOLUTIONS FOR WIRELESS DEVICES: ANALYSIS AND DEVELOPMENT

SUMMARY

In this project multi-presentational e-commerce solution prototype (e-shop) was designed and implemented. It has capabilities of working on multiple presentation areas simultaneously, including areas where very limited presentation capabilities are present, for example WAP/WML. Solution prototype is accessible via different devices – desktop, palm computers, and even low-end cellular phones.

Additionally, WAP protocol and problem domain research was accomplished, and possible e-commerce solution adoption methodology for WAP scene was presented.

In brief, following results were achieved:

1. Proved that it is possible to transform average-complexity e-commerce solutions from Web into low-capability wireless devices, using WAP/WML technologies (although some complex functionality might be lost)
2. Web e-commerce solution adoption methodology for WAP scene was introduced, including:
    - list of possible architectures for bringing solutions from Web to wireless, including pros and cons of each architecture
    - expandable rules for converting HTML to WML
3. Multi-presentational e-commerce solution prototype was created

1 ĮVADAS

Dažniausiai sutinkama elektroninės komercijos apraiška pasauliniame interneto www tinkle – elektroninės parduotuvės. Tačiau ši rinka jau perpildyta, o naujovių siūloma vis mažiau.

Šiuo metu itin sparčiai vystosi delninių kompiuterių paklausa – jų galimybės nuolat didėja, o kainos mažėja, todėl vis daugiau vartotojų iškeičia nešiojamąjį kompiuterį į delninį. Dauguma šių kompiuterių turi GPRS arba HSCSD duomenų perdavimo GSM tinklais galimybę, o kai kurie – net WLAN palaikymą. Šiais delniniais kompiuteriais galima prisijungti prie pasaulinio interneto tinklo ir naudotis jo teikiamomis galimybėmis. Dažnai tai daroma WAP protokolu, naudojant WML kalbą. Šias technologijas palaiko ir dauguma naujų mobiliųjų telefonų.

Šiame darbe bus bandoma sukurti elektroninės parduotuvės turinčios daugialypę prieigą prototipą, t.y. sprendimas turės prieigą tiek įprastiems ir delniniams kompiuteriams, tiek ir mobiliesiems telefonams. Todėl šiame darbe bus sprendžiami tokie uždaviniai:

- Atlikti esamų sprendimų analizę.
- Ištirti galimybę el. parduotuvės kūrimui naudoti:
  - universalų bei itin paplitusį WAP protokolą,
  - WML (*Wireless Markup Language*) kalbą.
- Šios dvi technologijos suteiktų galimybę el. parduotuvę pasiekti ir iš kitų įrenginių, tokių kaip mobiliųjų telefonų, turinčių WAP naršykles:
  - nepavykus panaudoti šių technologijų, pasirinkti kitas, geriausiai atitinkančias reikalavimus,
  - sėkmingo šių technologijų pritaikymo atveju – atlikti galimų būdų, kaip šias technologijas pritaikyti esamiems ir naujai kuriamiems elektroninės komercijos sprendimams analizę, bei atlikti pagrindinių jų privalumų bei trūkumų tyrimą.
- Sukurti el. parduotuvės veikiančios mobiliuosiuose įrenginiuose versiją.
- Suderinti veikimą su pasaulinio žiniatinklio el. parduotuvės programine įranga.

1.1 Tyrimo sritis, objektas ir problema

Šiame darbe bus tiriamos galimybės teikti elektroninės komercijos paslaugas mažo pajėgumo belaidžiams įrenginiams. Taip pat bus ieškoma metodų, kaip išplėsti esamus *WWW* žiniatinklio elektroninės komercijos sprendimus į belaidę sritį naudojant WAP technologijas, ir atsižvelgiant į belaidžių įrenginių bei tinklų įvedamus apribojimus. Taip pat bus ieškoma būdų kaip sukurti daugialypę prieigą turinčio elektroninės komercijos sprendimo prototipą. Šis sprendimas turi galėti dirbti keliose atvaizdavimo srityse tuo pačiu metu.

Kadangi dažniausiai pasitaikanti elektroninės komercijos rūšis yra elektroninės parduotuvės – prototipo kūrimui naudosime būtent ją.

WAP technologijos apibrėžia alternatyvą pasauliniam žiniatinkliui, ir kitiems turinio nešantiesiems tinklams. Tačiau WAP modelis buvo optimizuotas atsižvelgiant į itin didelius apribojimus, kurie labai skiriasi nuo šiandienos žiniatinkliui taikomų apribojimų:

- Belaidžiai įrenginiai, tokie kaip mobilieji telefonai, turi palyginus mažus ekranus
- Belaidžiai įrenginiai paprastai turi ribotas įvesties galimybes (neturi pelės ir turi ribotą klaviatūrą, su kuria paprastai dirbama tik viena ranka)
- Belaidžiai įrenginiai paprastai turi mažai apdorojimo galios ir atmintinės.
- Belaidžiai tinklai turi polinkį į ryšio praradimą, arba jo suprastėjimą
- Belaidžiai tinklai neretai turi labai ribotą pralaidumą (pvz. 9600 bps) ir didelį vėlinimą (laikas tarp užklausos ir atsakymo)
- Ir kt.

Todėl norint transformuoti elektroninės komercijos sprendimus iš *WWW* žiniatinklio į belaidę sritį, arba kuriant naujus elektroninės komercijos sprendimus WAP, būtina ištirti ar tai iš viso įmanoma, o jei taip - gerai suprasti ir įvertinti šiuos belaidės srities apribojimus, ir galimas jų pasekmes.

Pagrindinės šio tyrimo objekto pasirinkimo priežastys yra šios:

- El. parduotuvės – dažniausiai pasitaikanti el. komercijos sritis.
- Vis didesnis portatyviųjų įrenginių turinčių mikronaršykles paplitimas
- Labiausiai paplitusios ir daugiausiai įrenginių palaikomos yra būtent WAP/WML technologijos.

Norint pasiekti užsibrėžtus tikslus, iš pradžių reikės atlikti išsamią elektroninės komercijos bei WAP technologijų analizę, o taip pat pasirinkti pagalbines technologijas ir priemones su kuriomis dirbsime. Tačiau apie tai išsamiau parašyta analizės dalyje, skyriuje 2 („Elektroninės komercijos ir WAP analizė“, puslapis 11).

2 ELEKTRONINĖS KOMERCIJOS IR WAP ANALIZĖ

Pirmojoje analizės dalyje 2.1 („Elektroninės komercijos analizė“, puslapis 12), apžvelgsime atliktą išsamią elektroninės komercijos analizę:

- apibrėšime elektroninės komercijos sąvoką, bei trumpai aprašysime pagrindinius jos bruožus;
- aprašysime elektroninės komercijos kategorijas, bei panagrinėsime kiekvieną iš jų atskirai;
- išanalizuosime kokią įtaką elektroninė komercija daro verslui, įmonėms bei visuomenei;
- išsiaiškinsime kokia gali būti elektroninės komercijos veikla bei jos kryptys, kokioms sferoms gali būti taikomi elektroninės komercijos sprendimai, ir kokią naudą ji teikia kiekvienu atveju;
- išsamiau ištirsime kokią naudą iš elektroninės komercijos gali turėti abi pusės, t.y. jos „tiekėjai“ bei „vartotojai“;
- atliksime elektroninės komercijos pavyzdžių Lietuvoje analizę;
- išsamiau panagrinėsime tipinį elektroninės komercijos atvejį – elektroninė parduotuvę, ir sudarysime apibendrintą panaudojimo atvejų modelį, bei kitus iš jo sekančius panaudojimo atvejų modelius;
- apibendrinsime surinktą informaciją apie elektroninės komercijos sprendimus;

Antroje analizės dalyje 2.2 („WAP protokolo galimybių analizė“, puslapis 15) bus atliekama technologijos WAP, kurią numatome naudoti kaip elektroninės komercijos prieigą mobiliesiems įrenginiams, išsamios analizės apžvalga. Taip pat panagrinėsime pasirinktos technologijos privalumus ir trūkumus, bei palyginsime ją su kitomis technologijomis kurias būtų galima panaudoti šiam darbui.

Trečioje analizės dalyje 2.3 („Analizės bei projektavimo priemonių parinkimas“, puslapis 26) apžvengsime esamas, ir pasirinksime labiausiai tinkančias analizės bei projektavimo priemones.

Ketvirtojoje analizės dalyje 2.4 („Programinės įrangos, technologijų bei architektūros pasirinkimas“, puslapis 27) pasirinksime:

- Programos bendrąjį architektūrinį principą
- Programavimo kalbą
- Duomenų bazių valdymo sistemą
- Įrankius sekančioms užduotims atlikti:
  - Projektavimui
  - Programavimui
  - Dokumentacijos ruošimui

Šio, elektroninės komercijos bei WAP technologijos analizės, skyriaus pabaigoje padarysime bendras analizės dalies išvadas atsižvengdami į informaciją surinktą šiame skyriuje.

2.1 Elektroninės komercijos analizė

Šiame skyriuje atliksime trumpą darbo aplinkos – elektroninės komercijos apžvalgą. Norint atlikti šią apžvalgą, reikėjo atlikti išsamią analizę, kurią galima rasti prieduose – skyriuje 8.2 („Elektroninės komercijos analizė“, puslapis 140).

2.1.1 Elektroninės komercijos samprata

Elektroninę komerciją galima būtų apibrėžti kaip verslo formą, kada šalys bendrauja elektroniniu būdu, be fizinio ryšio. Elektroninė komercija - bendra sąvoka, aprėpianti verslo sandorius, valdomus elektroniniu būdu, naudojant telekomunikacijų tinklus.

Elektroninė komercija yra kurianti, vadovaujanti ir plečianti komercinius santykius internetu. Šis naujas verslas pasižymi sparčiai besiplečiančiomis pasiūlos galimybėmis, didėjančia visuotine konkurencija bei milžiniškais vartotojų lūkesčiais. Visame pasaulyje verslas keičia savo organizacines struktūras bei operacines formas: sena hierarchija pamažu nyksta, mažėja barjerų tarp įmonės klientų ir tiekėjų. Kad būtų įveiktos įsisenėjusios kliūtys, verslo procesai yra reorganizuojami, o į pačią reorganizaciją dažnai įtraukiama visa įmonė, jos partneriai, klientai ir net tiekėjai. Elektroninė komercija yra priemonė sudaryti sąlygas tokiems pasikeitimams bei juos paremti pasauliniu mastu. Ji leidžia įmonėms efektyviau ir lanksčiau atlikti vidaus operacijas, artimiau dirbti su tiekėjais bei jautriau reaguoti į klientų poreikius ir lūkesčius.

Sėkmingi elektroninės komercijos sumanymai gali apimti pirkimus, plėtrą ir produktų projektavimą, vadovavimą produkcijai ar gamybos rinkodarą, pardavimus, aptarnavimą, bendradarbiavimą versle, produktų platinimą, mokslinius tyrimus, informacijos skleidimą, komercinių bendruomenių steigimą, mokymą, renginius ir dar daug kitų verslo sferų.

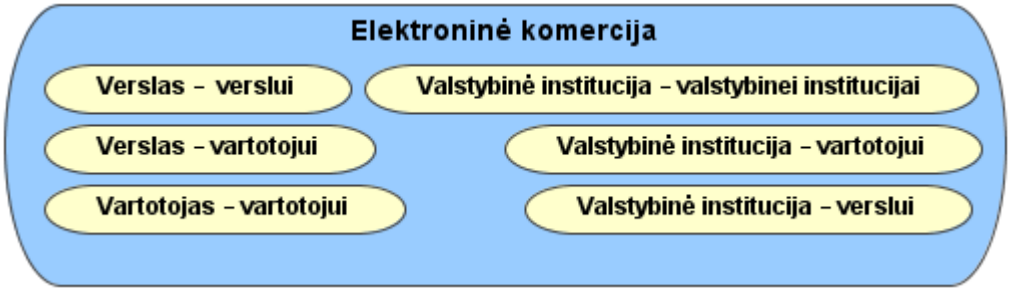
Vienas iš elektroninės komercijos atvejų būtų elektroninė prekyba. Elektroninę prekybą galima skaidyti į didmeninę, kai tiekiamas didelis prekių ar paslaugų užsakymas, ar mažmeninę, kai klientas dažniausiai yra tiesioginis vartotojas. Vis dėlto, nors šie specialūs atvejai yra didelės ekonominės svarbos, jie yra tik bendro elektroninio verslo operacijų modelio pavyzdžiai. Kiti nemažiau svarbūs pavyzdžiai galėtų būti įmonės vidinės transakcijos arba informacijos keitimasis tarp įmonių.

Elektroninės komercijos sprendimai gali būti įgyvendinti įvairiais lygiais - nuo paprasčiausio įmonės įjungimo į elektroninį tinklą iki sudėtingų elektroninių verslo procesų palaikymo.

2.1.2 Elektroninės komercijos formos, teikiama nauda

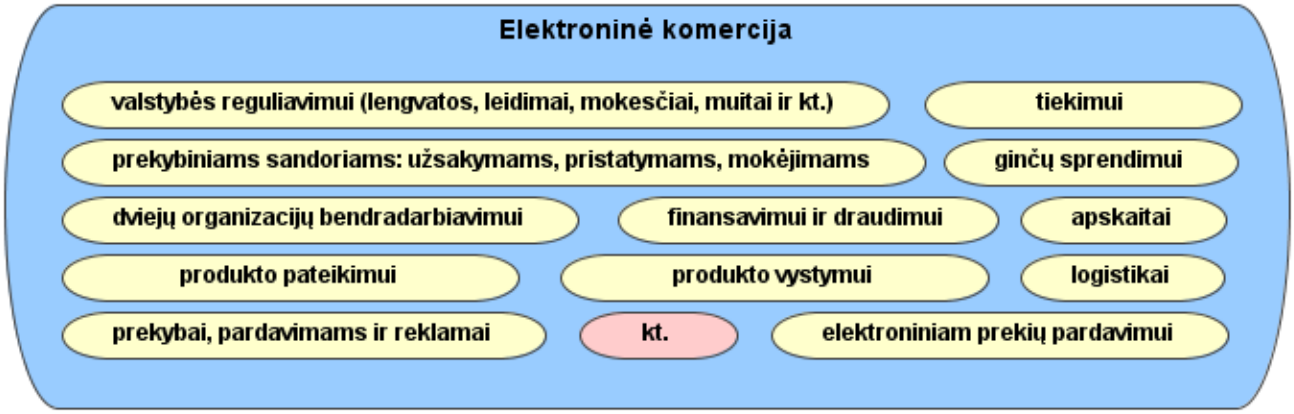
Elektroninė komercija gali apimti įvairias informacijos judėjimo bei sąveikos sferas. Norint įžvelgti plačias elektroninių priemonių taikymo galimybes, pravartu į elektroninę komerciją pasižiūrėti keliais aspektais.

Taigi pagal elektroniniu būdu bendraujančias šalis elektroninę komerciją galima skirstyti į kategorijas matomas sekančiame paveiksle.



2.1 pav. Elektroninės komercijos skirstymas pagal bendraujančias šalis

Taip pat elektroninės komercijos sprendimus galime skirstyti ir pagal taikymo verslo sferą, pavyzdys pateiktas sekančiame paveiksle.

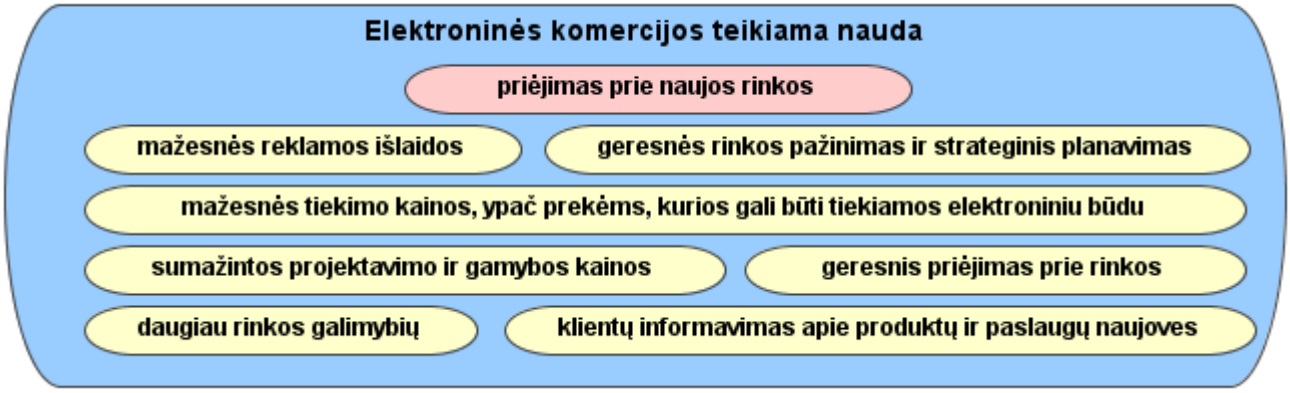


2.2 pav. Elektroninės komercijos skirstymas pagal taikymo sferą

Elektroninė komercija padeda atrasti naujas rinkas, naujas galimybes mažoms ir didelėms įmonėms pasauliniu mastu. Elektroninės komercijos tikslai yra padėti kompanijoms sukurti stipresnius ryšius su vartotojais bei verslo partneriais, nes geri santykiai su vartotojais bei pardavimo partneriais yra verslo sėkmė.

Elektroninės komercijos pranašumai yra mažesnės kainos, geriau pasiekiami vartotojai, spartesni atsakymai, didesnis pasirinkimas. Internetas yra geras susisiekimo kanalas - jis yra greitas, patikimas, nebrangus ir plačiai prieinamas, taip stiprinantis santykius su klientais bei partneriais.

Sekančioje iliustracijoje pateiksime pagrindinius ir dažniausiai sutinkamus elektroninės komercijos teikiamus privalumus.



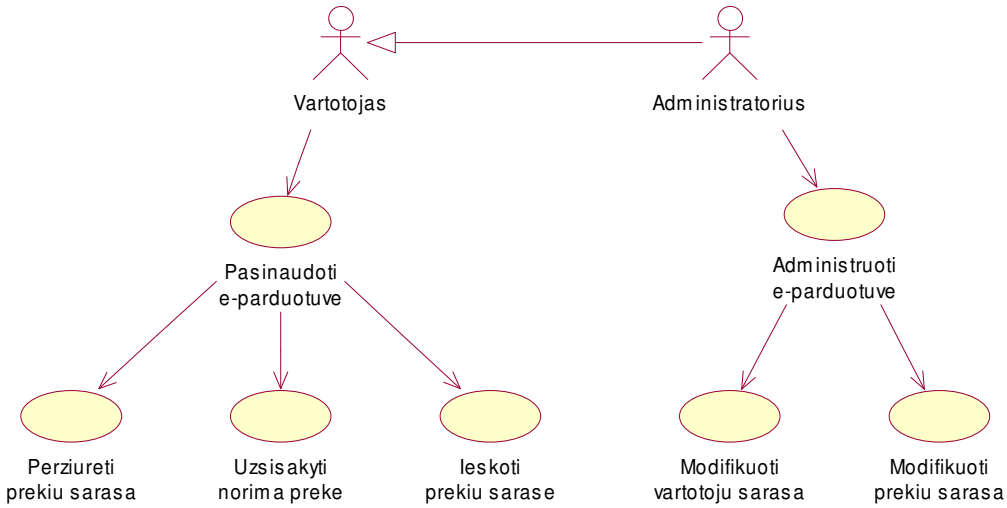
2.3 pav. Elektroninės komercijos teikiama nauda

Išsamiau elektroninės komercijos tipai, teikiama nauda, bei konkretūs pavyzdžiai Lietuvoje yra nagrinėjami skyriuje 8.2 („Elektroninės komercijos analizė“, puslapis 140).

2.1.3 Tipinis elektroninės komercijos atvejis - elektroninė parduotuvė

Dažniausiai sutinkama elektroninės komercijos apraiška pasauliniame interneto tinkle – elektroninė parduotuvė. Tačiau ši rinka jau perpildyta, o naujovių siūloma vis mažiau. Todėl šiame darbe bus bandoma išplėsti elektroninės parduotuvės maksimalios aprėpties sritį – sukurti elektroninės parduotuvės turinčios daugialypę prieigą prototipą, t.y. sprendimas turės prieigą tiek įprastiems ir delniniams kompiuteriams, tiek ir mobiliesiems telefonams.

Žemiau pateiksime keletą panaudojimo atvejų modelių, siekiant apibendrintai supažindinti su įprastais funkciniais elektroninės parduotuvės reikalavimais. Kadangi pateikiamas tik apibendrintas atvejis, jis nebūtinai sutampa su kuriamu prototipu.



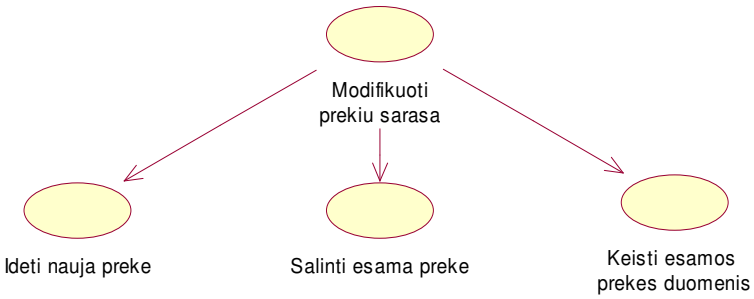
2.4 pav. Bendrasis panaudojimo atvejų modelis

Čia pagrindiniai aktoriai yra:

- Vartotojas – informacinėje sistemoje užregistruotas asmuo, galintis peržiūrėti prekių sąrašą, atlikti jame paiešką, bei užsakyt prekes.

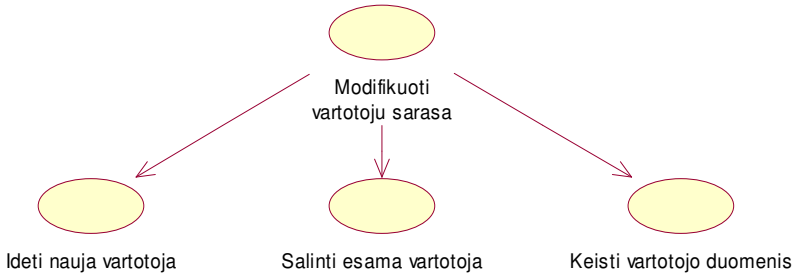
- Administratorius – tai informacinėje sistemoje užregistruotas asmuo, galintis modifikuoti prekių bei vartotojų sąrašus, t.y. keisti esamus, juos šalinti arba sukurti naujus.

Sekančiame paveiksle pateikiamas prekių sąrašo modifikavimo modelis (modifikuoti prekių sąrašą gali tik atitinkamas teises turintis asmuo, paprastai – administratorius).



2.5 pav. Prekių sąrašo modifikavimo modelis

Analogiškai atrodo ir vartotojų sąrašo modifikavimo modelis.



2.6 pav. Vartotojų sąrašo modifikavimo modelis

Pagal sudarytus elektroninės parduotuvės panaudojimo atvejų modelius, galime daryti išvadą jog jie atitinka standartinės elektroninės parduotuvės stereotipą. Kuriant panaudojimo atvejus, ir tuo pačiu ir būsimus funkcinius reikalavimus, nebuvo buvo atsižvelgta į tai, jog elektroninė parduotuvė turės veikti ir WAP/WML terpėje, bei atitikti šių technologijų apribojimus. Tačiau dėl apibendrinimo, buvo taikomas nesudėtingas modelis, todėl galima tikėtis jog daugumą aprašytų panaudojimo atvejų bus įmanoma realizuoti abejose darbo terpėse – tiek HTTP/HTML, tiek WAP/WML. Tai reiškia jog bus įmanomas elektroninės parduotuvės pasiekimas bei naudojimas iš daugumos mobiliųjų įrenginių, įskaitant mobiliuosius telefonus.

2.2 WAP protokolo galimybių analizė

Šiame skyriuje atliksime WAP protokolo galimybių apžvalgą. Norint atlikti šią apžvalgą, reikėjo atlikti išsamią analizę, kurią galima rasti prieduose – skyriuje 8.1 („WAP protokolo galimybių analizė“, puslapis 76).

WAP (angl. *Wireless Application Protocol*) [7] yra didelis žingsnis į priekį, suteikiantis universalią, internetu pagrįstą informacijos prieigą belaidžiams įrenginiams. Jis leidžia programinės

įrangos kūrėjams kurti vieną kartą – visiems pasaulio mobiliesiems tinklams. Nešėjai, pavyzdžiui, šiuo metu labiausiai paplitęs GSM, gali sukurti vartus (angl. *gateway*), kurie, panaudojant įvairius telefonus, leidžia naudotis visomis tinklo taikomosiomis programomis ir turiniu. Todėl, telefonų bei delninių kompiuterių gamintojai gali gaminti nebrangius įrenginius visiems nešėjams, nes juose naudojamas tas pats WAP protokolas.

WAP paskirtis – suteikti operatoriams, techninės įrangos gamintojams, programinės įrangos ir turinio kūrėjams bendrą aplinką, kuri leis kurti bei vystyti pridėtinės vertes paslaugas mobiliesiems įrenginiams. Keturi WAP įkūrėjai (Ericsson, Motorola, Nokia ir Unwired planet) kartu su kitais partneriais stengiasi sukurti globalias belaidžių paslaugų specifikacijas, nusakančias griežtus standartus kurie yra nepriklausomi nuo nešančiojo tinklo, bei jo struktūros. Visos protokolo savybės yra panaudojamos nepriklausomai nuo nešančiojo tinklo, bei įrenginių tipų. Organizacija kuri rūpinasi WAP standartais, bei yra jų savininkė (dabar jau valdoma *Open Mobile Alliance*) yra „WAP Forum“.

Mobilioji rinka sparčiai auga ir vartotojams siūlo naujas paslaugas. Norint mobiliesiems operatoriams ir įrangos gamintojams suteikti galimybę nugalėti sunkumus atsirandančius pažangių, įvairialypių, greitų ir lanksčių paslaugų teikime bei kūrime, WAP parenka bei aprašo aibę atvirų, išplečiamų protokolų bei turinio formatų, kaip šių įvairialypių (ir įrangos prasme) paslaugų įgyvendinimo pagrindą.

WAP Forum tikslai yra šie:

- Interneto turinį bei pažangias duomenų paslaugas padaryti pasiekiamas naudojant mobiliuosius telefonus ir kitus belaidžius terminalus.
- Sukurti globalią belaidžio protokolo specifikaciją, kuri galės veikti įvairių technologijų belaidžiuose tinkluose.
- Įgalinti turinio ir taikomųjų programų, kurios gerai veiktų labai plačiame nešančiųjų tinklų bei „galutinio vartotojo“ įrenginių spektre, kūrimą.
- Priimti ir išplėsti standartus bei technologijas kai tik atsiras poreikis.

WAP architektūros specifikacija pristato sistemą ir architektūras, kurios būtinos norint pasiekti WAP Forum užsibrėžtus tikslus. WAP architektūros specifikacija yra pradinis taškas siekiant suprasti WAP technologijas ir visas kitas iš jos sekančias specifikacijas. Ši specifikacija apžvelgia skirtingas naudojamas technologijas ir turi nuorodas į tolesnes bei detalesnes specifikacijas.

Dabartinė WAP architektūros specifikacija tęsia pirminės specifikacijos kryptį ir pasisekimą. Tinklo elementų funkcionalumas lieka panašus. Pavyzdžiui, architektūra naudoja našumo didinimo ir funkcionalumo gerinimo tarpinius serverius (*proxies*) tam, kad sumažintų apdorojimo reikalavimus apribotiems (greičiu, atminties kiekiu ir kt. parametrais) įrenginiams, atskleistų belaidžio tinklo galimybes ir funkcijas, bei suteiktų tinklo ir paslaugų valdymo galimybes. Dabartinė WAP

architektūros versija buvo pagerinta ir suteikia platesnį prisijungimo kelių pasirinkimą tarp klientų bei serverių, kai to reikia, pavyzdžiui, sutiekiant baigtis-baigtis (angl. *end-to-end*) saugumą.

WAP architektūros specifikacija suteikia karkasą plačiam protokolų, galimybių ir paslaugų spektrui. Tačiau ji yra informacinė, kadangi nenurodo jokių įgyvendinimo detalių ar specifikos.

2.2.1 WAP technologijos apžvalga

WAP sujungia tris technologijas: belaidį duomenų perdavimą, telefoniją ir internetą.

Tiek belaidžių duomenų perdavimo rinka, tiek internetas sparčiai auga ir pasiekia vis naujus vartotojus. Staigus interneto „augimas“ paspartino naujų ir įdomių informacinių paslaugų kūrimą. Dauguma naujai kuriamų internetinių technologijų yra skirtos staliniams ar galingesniems kompiuteriams, bei vidutinio arba didelio pralaidumo patikimiems duomenų tinklams. Masiškai parduodami maži belaidžiai įrenginiai (pavyzdžiui, mobilieji telefonai, delniniai kompiuteriai) turi labiau apribotus resursus, lyginant su įprastais kompiuteriais. Dėl šių esminių energijos, dydžio ir įperkamumo apribojimų mobilieji įrenginiai dažniausiai turi:

- Mažesnio galingumo procesorius
- Mažiau atminties (RAM ir ROM)
- Ribojamą energijos suvartojimą
- Mažesnius ekranus
- Skirtingus įvesties įrenginius (pvz. mobiliojo telefono mygtukus)

Panaši situacija yra ir su belaidžiais tinklais – komunikacijos aplinka labiau apribota lyginant su paprastais tinklais, todėl belaidžiai tinklai dažniausiai turi:

- Mažesnį pralaidumą
- Didesnį vėlinimą
- Mažesnį sujungimo stabilumą
- Mažiau numatomą buvimą (*availability*)

Norint suteikti papildomos vertės paslaugas mobilieji tinklai tampa sudėtingesni. Kad mobilieji tinklai atitiktų tokių tinklų operatoriams taikomus reikalavimus, jie turi būti:

- Suderinami – įvairių gamintojų terminalai (nesvarbu ar tai mobilus telefonas, ar delninis kompiuteris, ar kitas įrenginys) naudojasi mobiliojo tinklo paslaugomis
- Pritaikomi – mobiliųjų tinklų operatoriai gali pritaikyti paslaugas pagal klientų norus
- Efektyvūs – paslaugų kokybė pritaikyta mobiliojo tinklo veikimui ir charakteristikoms
- Patikimi – tieka pastovų ir numatomą pagrindą paslaugų tiekimui
- Saugūs – turi būti galimybė perduoti duomenis per neapsaugotus mobiliuosius tinklus, tačiau vartotojo duomenys turi išlikti nepakitę; taip pat įrenginiai ir paslaugos turi būti apsaugoti nuo saugumo problemų, pavyzdžiui, konfidencialumo praradimas

Dauguma dabartinių mobiliųjų tinklų turi pažangias paslaugas, kurios teikiamos nesudėtingu ir vartotojams patraukliu būdu, siekiant, kad šiomis paslaugomis naudotųsi kuo daugiau vartotojų. Standartinės galimybės (pavyzdžiui, skambučių valdymas) gali būti pagerintos naudojant WAP technologiją (pavyzdžiui, sukūrus vartotojo sąsają).

Pagrindinė belaidžių įrenginių savybė – mobilumas. Jis suteikia naujų galimybių paslaugoms, kurios gali teikti informaciją priklausančią nuo buvimo vietos.

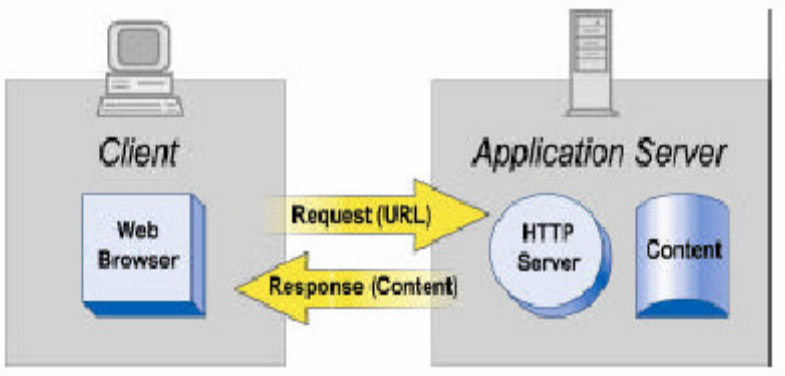
Taigi, WAP specifikacijos pritaikytos plačiam įrenginių spektrui: nuo įrenginių, kurie atlieka paprastas funkcijas (pavyzdžiui, mobilieji telefonai), iki įrenginių, kurių galimybės nuolat plečiasi (pavyzdžiui, delniniai kompiuteriai). Ši architektūra leidžia perkelti funkcionalumą į skirtingas tinklo dalis – t.y. arba į įrenginius, arba į tinklo darbines stotis, jei tokia būtinybė yra.

Taigi, be anksčiau jau paminėtų tikslų, WAP Forum dar iškėlė šiuos:

- Tiekti žiniatinkliu (*web*) pagrįstą taikymų modelį (angl. *application model*) belaidėms duomenų paslaugoms, naudojančioms telefoniją, mobilumą bei kitas unikalias belaidžių įrenginių ir tinklų savybes. Tačiau tuo pat metu palieka ir maksimalų lankstumą.
- Įgalinti įrenginio pritaikymą savo reikmėms (pavyzdžiui, tinkinti gaunamą turinį, jo pateikimą).
- Palaikyti saugų ir konfidencialų paslaugų teikimą bei komunikavimą, tuo pačiu išlaikant pastovumą ir atitikimą interneto saugumo modeliams.
- Palaikyti tiek dabartinius belaidžius įrenginius ir tinklus, tiek ir tuos kurie dar tik bus naudojami artimoje ateityje, įskaitant platų nešėjų spektrą – nuo siaurajuosčių iki plačiajuosčių.
- Tiekti saugų priėjimą prie vidinio įrenginio funkcionalumo.
- Palengvinti tinklo operatorių ir trečiųjų šalių teikiamų paslaugų tiekimą.
- Apibrėžti sluoksninę, pagal reikiamą mastą pritaikomą ir išplečiamą architektūrą.
- Kai tai įmanoma – įtakoti esamus ir besivystančius interneto standartus.

2.2.2 WAP architektūros apžvalga

WAP technologija remiasi pasaulinio interneto žiniatinklio sąvoka. Pasaulinio interneto žiniatinklio (*WWW*) architektūra suteikia labai lankstų ir galingą programavimo modelį. Taikomosios programos bei turinys yra vaizduojamas standartiniais duomenų formatais, peržiūrimais žiniatinklio naršyklėmis. Žiniatinklio naršyklė – tai tinklą naudojanti programa, kuri siunčia užklausas tinklo stotims, kad gautų įvardintus duomenų objektus, o tinklo serveris savo ruožtu atsako duomenimis, užkoduotais naudojant standartinius formatus.



2.7 pav. Pasaulinio žiniatinklio programavimo modelis

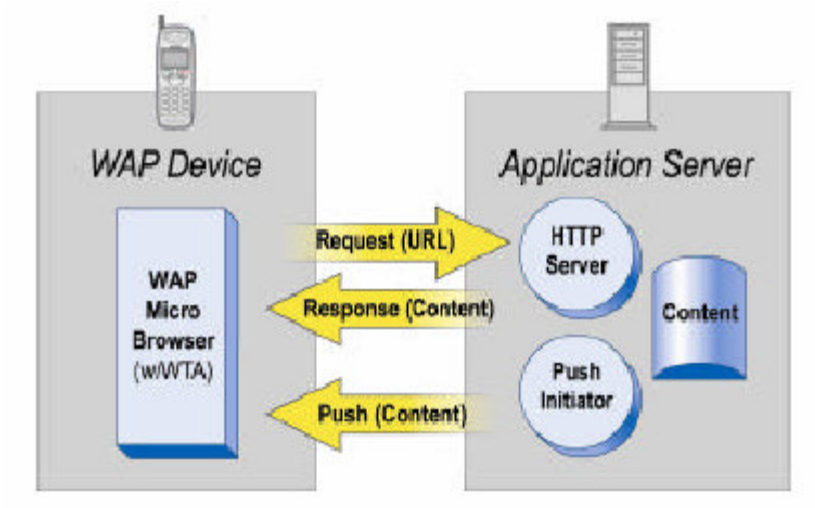
Pasaulinio žiniatinklio (*WWW*) standartai nurodo daug mechanizmų, kurie yra būtini kuriant bendros paskirties programavimo aplinką, pavyzdžiui:

- Standartinis pavadinimų modelis (*Standard naming model*) – visi serveriai ir visas pasaulinio žiniatinklio turinys yra pavadinti naudojant interneto standartą URL (vieningas resurso radėjas – *Uniform Resource Locator*) [12]
- Turinio tipizavimas (*Content typing*) – visas pasaulinio žiniatinklio turinys (tiksliau kiekvienas jo elementas) turi savo specifinį tipą, kuris žiniatinklio naršyklėms leidžia teisingai apdoroti turinį pagal jo tipą [13].
- Standartiniai turinio formatai (*Standard content formats*) – visos pasaulinio žiniatinklio naršyklės palaiko tam tikrą turinio formatų aibę (pavyzdžiui, HTML (*Hypertext Markup Language*) [4], scenarijų kalbos: *ECMAScript, JavaScript*).
- Standartiniai protokolai (*Standard Protocols*) – standartiniai tinklinio darbo protokolai leidžia bet kuriai žiniatinklio naršyklei komunikuoti su bet kuriuo žiniatinklio serveriu (pavyzdžiui, HTTP (*Hypertext Transport Protocol*) [11] protokolas veikiantis su TCP/IP protokolo rinkiniu. [10]).

Ši infrastruktūra leidžia vartotojams lengvai pasiekti didelį trečios-šalies programų ir turinio paslaugų skaičių. Taip pat ji leidžia programinės įrangos kūrėjams lengvai kurti programas bei turinio paslaugas plačiai klientų bendruomenei.

Patobulinus pasaulinio žiniatinklio programavimo modelį, gaunamas WAP programavimo modelis. Pasaulinio žiniatinklio programavimo modelio pritaikymas suteikia papildomos naudos programinės įrangos kūrėjų bendruomenei, kadangi, programavimo modelis jau pažįstamas, architektūra pasitvirtinusi ir kt. Atlikus tam tikrą optimizaciją ir pritaikius tam tikrus praplėtimus, WAP programavimo modelis pritaikytas belaidės aplinkos charakteristikoms. Pagrindiniai papildymai kurie buvo įdėti į WAP programavimo modelį yra:

- *Push*
- Telefonijos palaikymas (WTA)



2.8 pav. WAP programavimo modelis

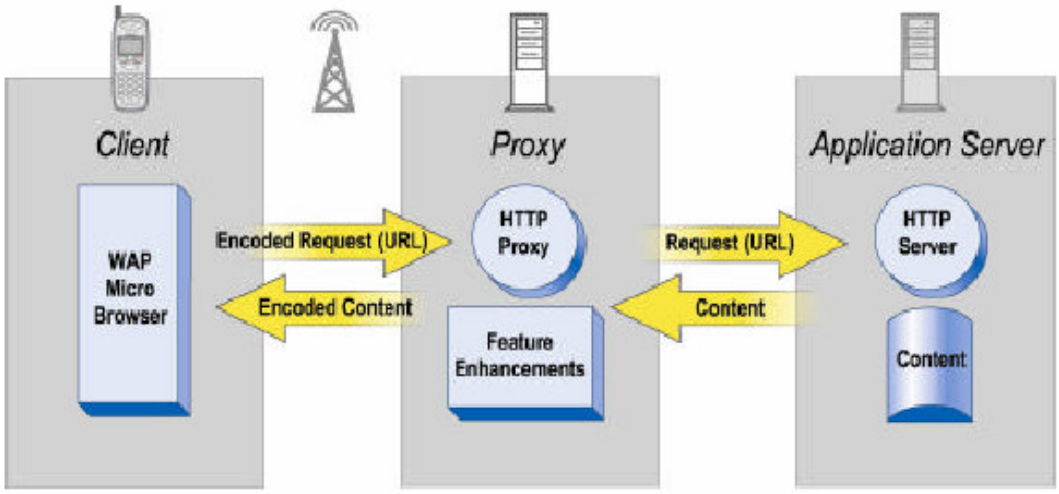
Klasikinis užklausos-atsakymo mechanizmas, dažnai dar vadinamas „*pull*“, skiriasi nuo „*push*“ mechanizmo. WAP turinys bei programos yra apibrėžtos aibe gerai žinomų turinio formatų, pagrįstų panašiais pasaulinio žiniatinklio (WWW) turinio formatais. Turinys yra transportuojamas naudojant standartinių komunikavimo protokolų aibę, kurie savo ruožtu taip pat yra pagrįsti panašiais pasaulinio žiniatinklio komunikavimo protokolais. WAP mikronaršyklė belaidžiame terminale koordinuoja vartotojo sąsają, ir yra analogas pasaulinio žiniatinklio naršyklei.

WAP apibrėžia standartinių komponentų aibę, kurie įgalina komunikavimą tarp mobiliųjų terminalų ir tinklo serverių.:

- Standartinis pavadinimų modelis (*Standard naming model*) – naudojamas pasaulinio žiniatinklio vieningas resurso radėjas (*Uniform Resource Locator - URL*) tam kad atpažintų WAP turinį serveriuose. Pasaulinio žiniatinklio vieningas resurso identifikatorius (*Uniform Resource Identificator - URI*) yra naudojamas identifikuojant vidinius įrenginio resursus, pavyzdžiui,. skambučių valdymo funkcijas [12].
- Turinio tipizavimas (*Content typing*) – visas WAP turinys (tiksliau kiekvienas jo elementas) turi savo specifinį tipą, taip leisdamas WAP naršyklėms teisingai apdoroti turinį pagal jo tipą. WAP tipai paprastai atitinka egzistuojančius pasaulinio žiniatinklio tipus [13].
- Standartiniai turinio formatai (*Standard content formats*) – WAP turinio formatai yra pagrįsti pasaulinio žiniatinklio technologija ir savyje turi atvaizdavimo žymių, kalendoriaus informacijos, elektroninių biznio kortelių objektus, paveikslus ir scenarijų kalbą.
- Standartiniai komunikavimo protokolai (*Standard communication protocols*) – WAP komunikavimo protokolai leidžia WAP naršyklei komunikuoti su žiniatinklio serveriu.

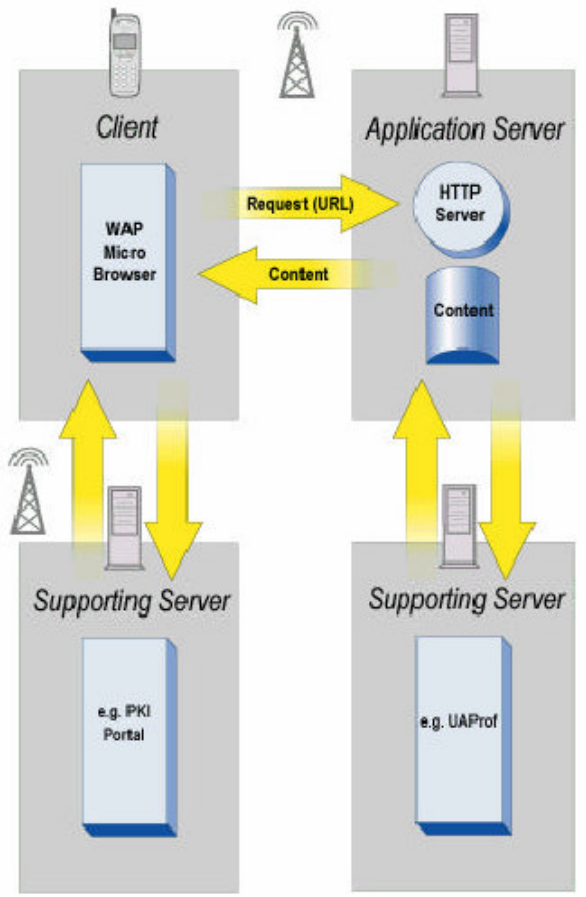
Siekiant pagerinti ir optimizuoti sujungimą tarp belaidės srities ir pasaulinio žiniatinklio, WAP naudoja tarpinio serverio technologiją, kuri gali atlikti įvairias funkcijas (pavyzdžiui, protokolo vartų

(*Protocol Gateway*), turinio užkodavimo ir dekodavimo, vartotojo agento profilio valdymo (*User Agent Profile Management*), spartinančiojo tarpinio serverio (*caching proxy*) funkcijas).



2.9 pav. Savybes bei našumą pagerinantys tarpiniai serveris

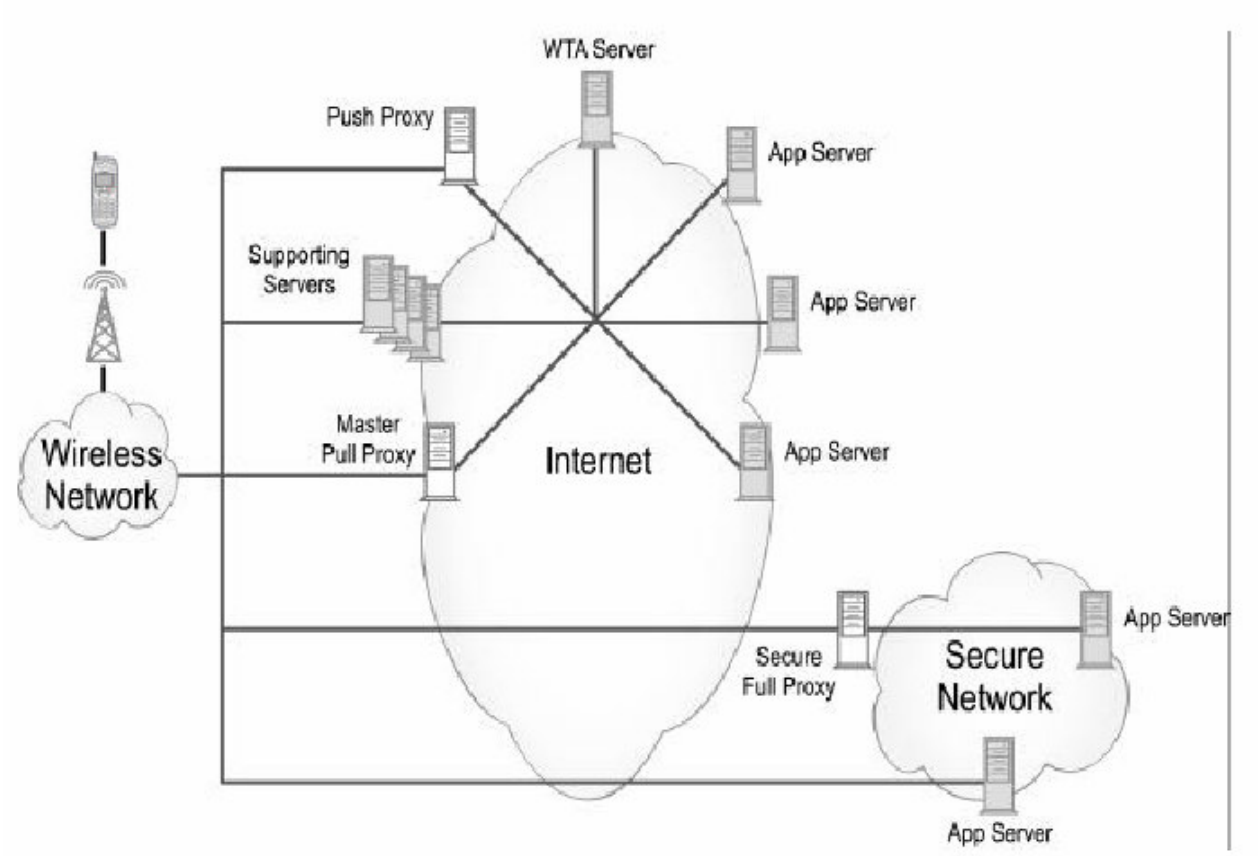
Ši infrastruktūra mobiliųjų terminalų vartotojams leidžia pasiekti įvairų interneto turinį ir taikomąsias programas, o programinės įrangos gamintojams – kurti vieningas turinio paslaugas ir taikomąsias programas.



2.10 pav. Pagalbiniai serveriai

WAP architektūroje taip pat yra pagalbiniai serveriai (pavyzdžiui, PKI Portal, UAProf Server, Provisioning Server), kurie teikia paslaugas įrenginiams, tarpiniams serveriams bei kai reikia – taikomajai programinei įrangai [7].

Dažniausiai pasitaikantis WAP vartojimas susideda iš pasaulinio žiniatinklio serverio, WAP spartinančiojo tarpinio serverio ir WAP kliento, tačiau WAP architektūra gali palaikyti ir kitas konfigūracijas.



2.11 pav. Pavyzdinė WAP tinklo schema

WAP klientai komunikuoja su taikomųjų programų serveriais per kelis skirtingus tarpinius serverius arba tiesiogiai. WAP klientai palaiko tarpinių serverių pasirinkimo mechanizmą kuris leidžia jiems naudoti labiausiai tinkamą tarpinį serverį duotai paslaugai arba esant reikalui – netgi jungtis tiesiai prie paslaugos. Tarpiniai serveriai gali būti naudojami užklausos papildymui. Jie „vertėjauja“ tarp WAP ir žiniatinklio protokolų (HTTP, TCP), taip leisdami WAP klientams siųsti užklausas reikiamam serveriui.

Tarpiniai serveriai gali būti skirtingose vietose, įskaitant belaidžius nešėjus arba nepriklausomus paslaugų tiekėjus, tam kad tiekti savybių pagerinimus surištus su belaidžiu tinklu, pavyzdžiui telefonija, vietos nustatymas, nešėjo paslaugų gavimas (angl. *provisioning*), arba optimizuoti komunikavimą tarp įrenginio ir taikomųjų programų serverio, pavyzdžiui protokolų vertimas, arba slapukai (angl. *cookies*), spartinimas (angl. *caching*). Tarpiniai serveriai gali būti saugiame tinkle, tam kad tiekti saugų komunikavimo kanalą tarp belaidžio įrenginio ir saugaus tinklo.

Kai kuriais atvejais įrenginiai gali jungtis tiesiogiai prie taikomųjų programų serverių, pavyzdžiui tam kad pasiekti maksimaliai saugų tiesioginį susijungimą tarp šių dviejų taškų (kliento – serverio).

Pagalbiniai serveriai teikia pagalbines funkcijas kurios yra būtinos arba naudingos įrenginiams, tarpiniams serveriams bei taikomųjų programų serveriams. Šios funkcijos gali būti aprūpinimas (angl. *provisioning*), PKI, vartotojų agentų profiliai ir kt.

Toliau apžvelgsime WAP įrenginių architektūrą.

2.12 pav. WAP kliento architektūra

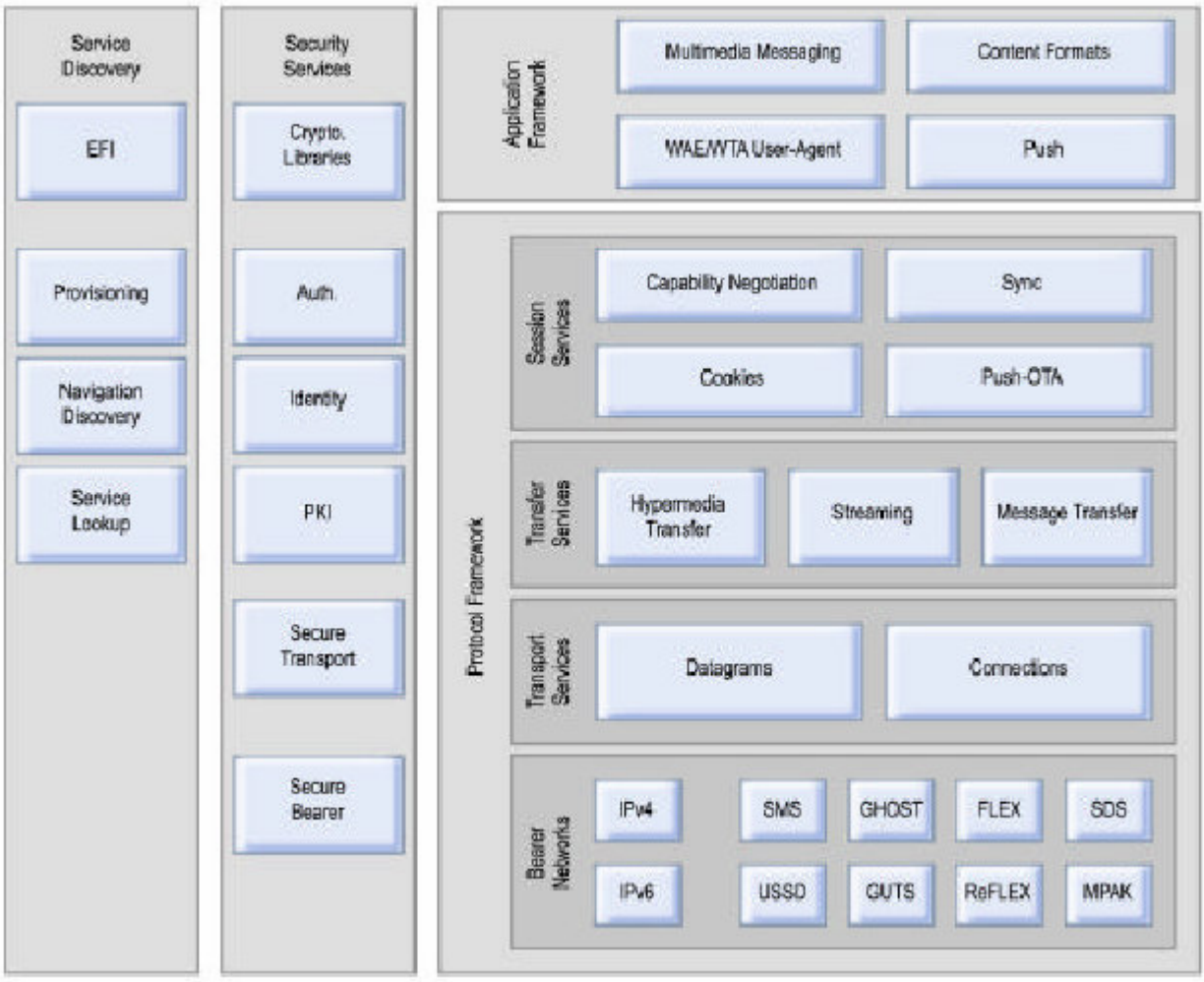
Taikomųjų programų karkasas (angl. *Application Framework*) įrenginiui teikia vykdymo aplinką WAP taikomosioms programoms. WAP taikomosios programos yra sudarytos iš žymų, scenarijaus, stiliaus lentelių bei daugialypės terpes turinio – visa tai yra perteikiama įrenginyje. WAP taikomųjų programų aplinkos WAE (angl. *WAP Application Environment*) apdorojimo modelis nurodo struktūrą, kurioje šios įvairios vykdomojo ir nevykdomojo turinio formos sąveikauja.

WAP tinklo protokolus naudoja klientas ir serveris. Turinio atvaizduotojai (angl. *renderers*) interpretuoja specifines turinio formas, ir jas pateikia galutiniam vartotojui, kad šis galėtu jas peržiūrėti bei atlikti norimus veiksmus. Bendros funkcijos yra nustatytos taip, kad būtų išnaudojamos taikomosios programos karkaso, įskaitant nuolatinį buvimą (angl. *persistance*) bei duomenų sinchronizavimą.

Belaidžio identifikavimo modulis WIM (angl. *Wireless Identity Module*), kaip pažymėta [7], turi savyje įrenginio tapatumą (angl. *identity*) bei kriptografines priemones abipusiam WAP įrenginių bei serverių tapatumo patvirtinimui.

Architektūra teikia mechanizmą prieiti prie išorinių funkcijų kurios yra įmontuotos arba pridėtuos prie įrenginių naudojant išorinio funkcionalumo sąsają (angl. *External Functionality Interface - EFI*).

WAP architektūra teikia keičiamos apimties bei išplečiamą taikomųjų programų kūrimo aplinką mobiliesiems komunikavimo įrenginiams. Tai yra pasiekiama naudojant sluoksnuotą protokolų dėklo architektūrą.



2.13 pav. WAP dėklo architektūra

Kiekvienas protokolų dėklo architektūros sluoksnis suteikia funkcijų ir / arba paslaugų aibę kitoms paslaugoms ir taikomosioms programoms per gerai apibrėžtų sąsajų aibę. Kiekvienas architektūros sluoksnis yra pasiekiamas aukštesniuose sluoksniuose, o taip pat paslaugoms bei taikomosioms programoms.

WAP architektūra atskiria paslaugų sąsajas nuo protokolų kurie teikia šias paslaugas.

2.2.3 WAP saugumas - WTLS

Sparčiai auganti belaidžio ryšio rinka didina mobiliųjų įrenginių, su papildoma verte / funkcijomis, poreikį. WAP yra sukurtas norint patenkinti šį poreikį. WAP aprašo protokolų aibę pažangių mobiliųjų paslaugų kūrimui. Kadangi ši technologija yra vystoma tarptautinės organizacijos (*WAP Forum*), tai leidžia mobiliojo ryšio vartotojui naudotis WAP teikiamomis paslaugomis visame pasaulyje, nepriklausomai nuo vietinės mobiliojo ryšio technologijos. WAP yra nepriklausoma nuo nešančiųjų paslaugų.

Todėl vis labiau svarbus tampa belaidžio transporto sluoksnio saugumo protokolas WTLS (angl. *Wireless Transport Layer Security*) yra WAP architektūros saugumo sluoksnis. Pirminiai jo tikslai yra privatumo, duomenų integralumo ir autentiškumo nustatymo teikimas WAP taikymams. Saugumas yra

reikalingas tam kad saugiau prisijungti prie tokių paslaugų kaip elektroninė komercija arba bankininkystė. Vartotojas nori būti tikras, kad naudojamos paslaugos yra būtent tos kurių jis tikisi ir kad šios paslaugos yra saugios. Tačiau mobilieji tinklai negali suteikti pilno baigties-baigties (angl. *end-to-end*) saugumo. Tai ir yra pagrindinė priežastis kodėl WTLS yra reikalingas.

WTLS teikia transporto paslaugų sąsajas aukštesniam sluoksniui. Ši sąsaja yra panaši į transporto paslaugų sąsajas esančias „žemiau“ WTLS. WTLS yra pagrįstas gerai žinomu TLS v1.0 saugumo sluoksniu, plačiai paplitusiu internete. Žinoma, dėl belaidžių tinklų prigimties, buvo reikalingos modifikacijos ir pakeitimai. Belaidžiai tinklai reikalauja tiek datagramomis, tiek ir susijungimais orientuoto transporto sluoksnio protokolų. Mobilioji įranga taip pat iškelia reikalavimus algoritmams, nes apdorojimo galia ir turimos atmintinės kiekis yra ribotas. Be to, reikia įvertinti mažą pralaidumą, bei apribojimus kriptografijos panaudojime ir eksportavime.

WTLS savyje turi naujų savybių: datagramų palaikymas, optimizuotas paketo dydis ir pasisveikinimas (angl. *handshake*), dinaminis raktų atnaujinimas. Jis buvo optimizuotas mažo pralaidumo nešantiesiems tinklams, turintiems palyginus didelį vėlinimą. Mobilioji įranga (mobilieji įrenginiai) gali būti sukonstruota taip, kad palaikytų tik tam tikrą šifravimo rinkinių aibę.

Nors belaidžiai tinklai ir sukuria daug reikalavimų – teikti priimtiną saugumo lygį įmanoma. Tačiau kita vertus, absoliutaus saugumo ir gero patogumo derinys net teoriškai nėra įmanomas. Tos saugumo problemos, kurios jau yra aptiktos WTLS, yra modifikacijų pritaikant protokolą belaidžiai sričiai, pasekmė. Laimei WTLS vis dar plėtojamas, ir visos problemos ištaisomos, suteikiant vis daugiau saugumo. Žinoma visapusis saugumas priklauso ir nuo kitų faktorių, ne tik nuo WTLS.

2.2.4 LEAP – alternatyva WAP

Lengvas ir efektyvus taikymų protokolas LEAP (angl. *Lightweight and Efficient Application Protocol*) yra bendras didelio našumo efektyvių protokolų karkasas, kurie yra idealūs belaidžiui bei mobiliam taikymui, aibei. LEAP yra suprojektuotas atsižvelgiant į visus belaidės duomenų komunikacijos industrijos reikalavimus, ir yra orientuotas į didžiausios naudos industrijai bei vartotojui teikimą.

LEAP yra belaidžio taikymo protokolų aibė, optimizuota mažų pranešimų perdavimui belaidžiais tinklais. Belaidžių tinklų galimybes riboja nedidelis pralaidumas, o nedidelių gabaritų įrenginių galimybes – tokie dalykai kaip ekrano dydis, baterijų bei atmintinės talpa. Šie apribojimai labai sureikšmina duomenų perdavimo efektyvumą.

LEAP protokolai yra apie penkis kartus efektyvesni nei įprastas SMTP elektroninių laiškų protokolas. Šis pagerintas efektyvumas reiškia ilgesnį baterijų darbo laiką mobiliesiems telefonams, delniniams kompiuteriams bei kitiems belaidžio interneto įrenginiams.

LEAP protokolų patentai nekainuoja, o atviro kodo protokolų įgyvendinimai jau yra sukurti įvairiems įrenginiams bei pranešimų centrų platformoms. Kadangi protokolai yra sukurti ir prieinami,

jie gali būti greitai išplatinti bei įgyvendinti kaip perspektyvi WAP alternatyva. Pagrindinės bendros WAP ir LEAP charakteristikos pateiktos lentelėje.

2.1 lentelė WAP ir LEAP palyginimas

WAP	LEAP
Reikia žinoti patentų apribojimus	Nėra patentų
Publikuotas savarankiškai, WAP Forum	Publikuotas kaip RFC
Versijos gali keistis be perspėjimo	Visos versijos yra fiksuotos
Prižiūrimas WAP Forum	Prižiūrimas atviros darbo grupės
Egzistuojančių protokolų „atradimas iš naujo“	Efektyvumą optimizuojantys išplėtimai egzistuojantiems protokolams
Priklausomas nuo įrenginio vartotojo sąsajos charakteristikų	Nepriklausomas nuo vartotojo sąsajos
Būdingos saugumo spragos	Nėra saugumo prielaidų
Nepastovus protokolo numerio priskyrimas	Pastovus protokolo numerio priskyrimas
Prastas techninis modelis	Geras techninis modelis
Pagrindinis tikslas: žiniatinklio naršymas	Pagrindinis tikslas: pranešimų siuntimas

Tačiau nors teoriškai LEAP ir turi daug privalumų lyginant su WAP (tai itin afišuoja LEAP kūrėjai), LEAP nėra toks paplitęs kaip WAP, o jo ateitis atrodo miglota.

Šiuo metu kitų žinomų bei apčiuopiamus rezultatus pasiekusių alternatyvų WAP nėra.

2.3 Analizės bei projektavimo priemonių parinkimas

Pagrindinė technologija naudojama programinės įrangos modeliavimui, specifikavimui, vizualizavimui bei dokumentavimui šiuo metu yra UML (*Unified Modeling Language*) [3]. Šiame darbe taip pat naudosime UML – dėl paprastumo, dokumentacijos bei įrankių gausos.

Šiuo metu galima išskirti 3 pagrindinius / stambiausius produktus skirtus UML CASE modeliavimui:

- Rational Rose (http://www.rational.com)
- MagicDraw UML (http://www.magicdraw.com)
- Microsoft Visio (http://office.microsoft.com/visio)

Žemiau lentelėje pateiktas šių produktų palyginimas:

2.2 lentelė UML CASE įrankų palyginimas

UML CASE įrankių pasirinkimo kriterijai	Rational Rose	MagicDraw	Microsoft Visio
Pilnas UML (1.4 versijos) palaikymas	+	+	+
Diagramų suderinamumo kontrolė	+	+	+
Diagramų pasirinkimo sąrašai	+	+	+
Modelio navigavimas	+	+	+

UML CASE įrankių pasirinkimo kriterijai	Rational Rose	MagicDraw	Microsoft Visio
Spausdinimas	+	+	+
Dokumentavimas HTML	+	+	+
Įrankio gyvybingumas (*robustness*)	+	+	+
Diagramų eksportas	+	+	+
Versijų palaikymas	+	+	+
Kodo generavimas	+	+	+
Duomenų modeliavimas	+	-	-
Tiesioginis ir atvirkštinis projektavimas (*Round-trip engineering*)	+	+	+
Repozitorijos palaikymas	+	+	+
RUP pagalbinis palaikymas	+	-	-

Kaip matome iš lentelės visų šių įrankių daugiau nei pakanka mūsų darbui atlikti. Šiam darbui atlikti pasirinksime Rational Rose, kadangi yra mažiau universalus ir labiau į probleminę sritį orientuotas įrankis nei Microsoft Visio, ir todėl kad šio produkto veikimui, priešingai nei MagicDraw, nėra reikalinga Java aplinka (*Java Runtime*).

2.4 Programinės įrangos, technologijų bei architektūros pasirinkimas

Norint pasiekti maksimalų suderinamumą su visais įrenginiais, įprastinė kliento-serverio architektūra, kur nemaža dalis logikos įgyvendinta kliento pusėje, čia netinkama. Todėl visa logika bus žiniatinklio serveryje, o kliento vaidmenį atliks kliento įrenginyje esanti naršyklė. Tai reiškia jog kuriamo elektroninės komercijos sprendimo veikimas turės būti pagrįstas kliento užklausomis, ir žiniatinklio serverio atsakymais į šias užklausas (atsakymai pateikiamai vartotojui siunčiant atitinkamos žymų kalbos dokumentus-puslapius, prieš tai atliekant reikiamus pakeitimus serveryje). Šis architektūros pasirinkimas yra sąmoningai atliekamas prieš pasirenkant technologijas, nes tik žinant architektūrą galima išsirinkti tinkamiausias technologijas jai įgyvendinti.

Elektroninei parduotuvei realizuoti pasirinkome PHP [1] programavimo kalbą. Šį sprendimą įtakojo daugelis PHP turimų teigiamų savybių:

- Itin gera dokumentacija
- PHP scenarijus yra lengvai įterpiamas į HTML/WML kodą, ir atvirkščiai - HTML/WML kodas yra lengvai įterpiamas į PHP scenarijus
- PHP turi standartinę sintaksę (pagrindai yra perimti iš C, Perl, Java)
- Veikia skirtingose operacinėse sistemose (OS) (Windows, Linux, Unix).
- Ji suderinama su beveik visais šiuolaikiniais žiniatinklio serveriais (Apache, IIS ir kt.)
- Platinama visiškai nemokamai

- Lengva įdiegti
- Lengva išmokti ir pritaikyti.
- Atviro kodo kalba, todėl jį tobulina didelė grupė žmonių
- Pasižymi dideliu greičiu serverio pusėje, bei dirbant su duomenų bazėmis
- Gausus pavyzdinių programų ir šablonų pasirinkimas

Elektroninei parduotuvei pasirinkome MySQL [2] duomenų bazių valdymo sistemą, nes:

- Platinama visiškai nemokamai
- Lengva įdiegti
- Itin gera dokumentacija
- Veikia skirtingose operacinėse sistemose (OS) (Windows, Linux, Unix)
- MySQL – dažniausiai su PHP naudojama duomenų bazių valdymo sistema

Programos kodo rašymui bus naudojamas nemokamas, itin greitas, mažos apimties, daug funkcijų turintis bei skirtingose operacinėse sistemose veikiantis tekstinis redaktorius SciTE (http://scintilla.sourceforge.net/SciTE.html)

Projektavimo įrankiu pasirinktas „*Rational Rose*“, todėl jis taip pat naudojamas ir ruošiant dokumentaciją. Dokumentacija rašoma naudojant Microsoft Word tekstinį redaktorių.

2.5 Analizės išvados

Atlikus elektroninės komercijos, vienos jos atšakų – elektroninės parduotuvės, bei WAP technologijų analizę galima daryti išvadą jog darbo tema pasirinkta teisingai, bei darbo atlikimas panaudojant WAP/WML technologijas kaip vieną iš nešančiųjų terpių yra įmanomas, o tuo pačiu ir elektroninės parduotuvės pasiekimas bei naudojimas iš daugumos mobiliųjų įrenginių, įskaitant mobiliuosius telefonus.

Pagal sudarytus elektroninės parduotuvės panaudojimo atvejų modelius, galime daryti išvadą jog jie atitinka standartinės elektroninės parduotuvės stereotipą. Kuriant panaudojimo atvejus, ir tuo pačiu ir būsimus funkcinius reikalavimus, nebuvo buvo atsižvelgta į tai, jog elektroninė parduotuvė turės veikti ir WAP/WML terpėje, bei atitikti šių technologijų apribojimus. Tačiau dėl apibendrinimo, buvo taikomas ne itin sudėtingas modelis, todėl galima tikėtis jog daugumą aprašytų panaudojimo atvejų bus įmanoma realizuoti abejose darbo terpėse – tiek HTTP/HTML, tiek WAP/WML. Tai reiškia jog bus įmanomas elektroninės parduotuvės pasiekimas bei naudojimas iš daugumos mobiliųjų įrenginių, įskaitant mobiliuosius telefonus.

Darbo atlikimui programavimo kalba PHP bei duomenų bazių valdymo sistema MySQL neigiamos įtakos neturės. PHP palengvins vartotojo sąsajos kodo generavimą, ir leis naudoti bendą programos kodo dalį, skirtą logikai, bei darbui su duomenų bazėmis. MySQL duomenų bazių valdymo

sistema, bei modeliavimo įrankis „*Rational Rose*“ pilnai atitinka visus šio darbo keliamus reikalavimus.

Apie WAP galima pasakyti jog tai yra dar palyginus jauna technologija, kurios trūkumai yra sparčiai šalinami. WAP įgauna vis daugiau funkcionalumo, išbaigtumo ir saugumo.

WAP apibrėžia alternatyvą pasauliniam žiniatinkliui, turinio nešantiesiems tinklams. WAP modelis buvo optimizuotas atsižvelgiant į apribojimus, kurie yra labai skirtingi lyginant su šiandienos žiniatinkliui taikomų apribojimų:

- Belaidžiai įrenginiai, tokie kaip mobilieji telefonai, turi palyginus mažus ekranus
- Belaidžiai įrenginiai paprastai neturi pelės ir turi ribotą klaviatūrą, kuri paprastai yra naudojama viena ranka
- Belaidžiai įrenginiai paprastai turi mažai apdorojimo galios ir atmintinės.
- Belaidžiai tinklai turi polinkį į ryšio praradimą, arba jo suprastėjimą
- Belaidžiai tinklai neretai turi labai ribotą pralaidumą (pvz. 9600 bps) ir didelį vėlinimą (laikas tarp užklausos ir atsakymo)
- Belaidžiai tinklai gali palaikyti arba nepalaikyti IP

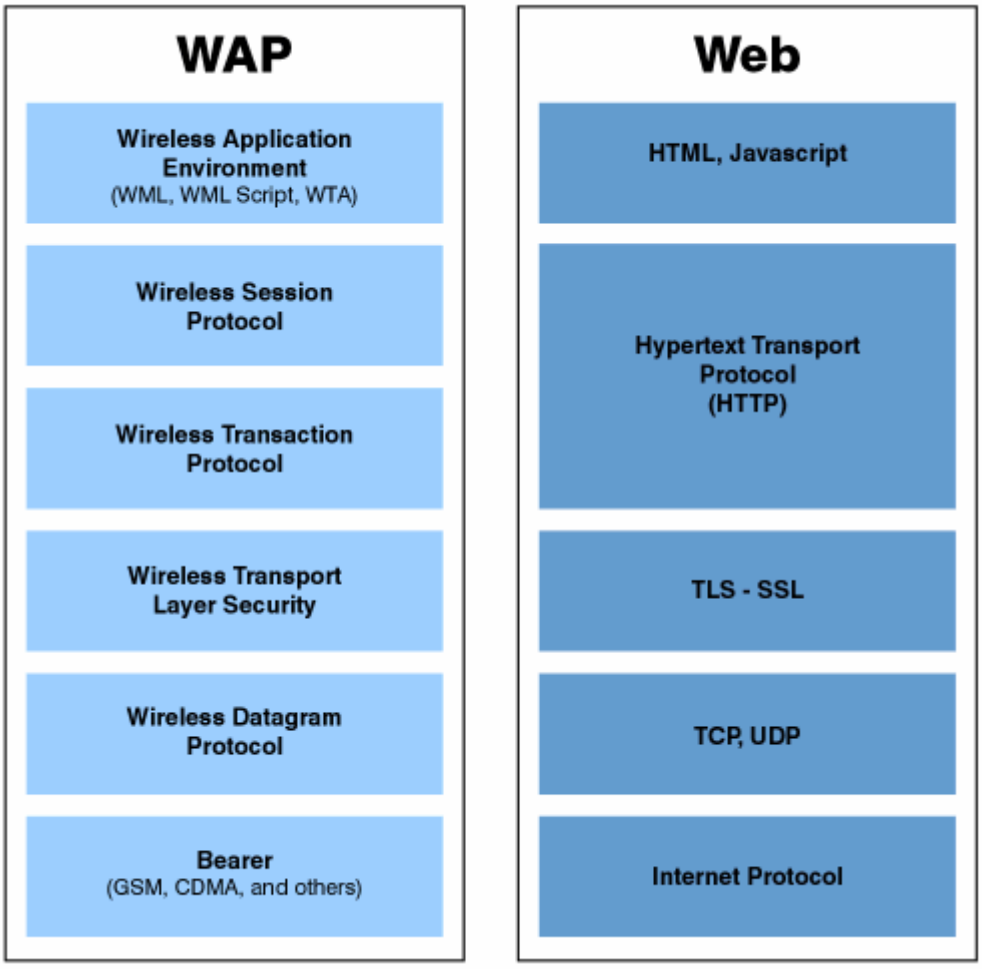
Kadangi belaidžiai įrenginiai darosi vis galingesni, tikėtinas variantas jog įmanoma tai, kad WAP ir žiniatinklio modeliai kada nors susilies į bendrą standartą. Taip pat įmanoma ir tai, jog modeliai išsiskirs, nes mažų, belaidžius tinklus naudojančiu įrenginių paskirtis gali išlikti specifinė. Kadangi WAP modelis yra pagrįstas sluoksnine architektūra, jis bus lengvai pritaikomas ateities reikmėms – kad ir kokios jos būtų.

3 ELEKTRONINĖS KOMERCIJOS SPRENDIMŲ TAIKYMO WAP TECHNOLOGIJOSE METODIKA

Daugumą elektroninės komercijos sprendimų galima pasiekti tik įprastomis HTML naršyklėmis. Naujos terpės atsiradimas elektroninės komercijos sprendimams yra labai svarbus – tai visada siejama su padidėjusiu sistemos vartotojų skaičiumi. Šioje darbo dalyje mes nagrinėsime elektroninės komercijos perkėlimo į WAP sritį problemą, o taip pat sukursime elektroninės komercijos sprendimo (elektroninės parduotuvės) prototipą, kurį bus galima pasiekti ir įprastomis HTML naršyklėmis, ir naudojant mobiliųjų įrenginių WML mikro naršykles. Tai taip pat bus ir pavyzdys kaip vienas sprendimas gali būti skirtas daugiau nei vienai terpei (nebūtinai HTML ir WML).

WAP yra alternatyvi pasauliniam WWW žiniatinkliui, duomenų pateikimo bei bendravimo su vartotoju terpė. WAP modelis buvo optimizuotas atsižvelgiant į didelius apribojimus (kurie atsiranda dėl belaidžio nešėjo bei mobiliųjų įrenginių) kurie labai skiriasi nuo žiniatinkliui taikomų apribojimų

Sekančioje schemoje pateikiamos WAP bei WWW žiniatinklyje naudojamos technologijos, o taip pat iš dalies ir kuri WWW žiniatinklio technologija yra kurios WAP technologijos analogas:



3.1 pav. Pagrindinės WAP ir WWW žiniatinklio technologijos

Keletas neįprastų faktų apie WAP:

- WWW žiniatinklis vartotojams suteikė galimybę pasiekti informaciją esančią bet kuriame pasaulio serveryje, o WAP taip pat suteikia ir galimybę tą informaciją pasiekti iš bet kurios vietos,
- didžioji dalis kritikos, tenkančios WAP, iš tikrųjų turėtų būti skirta ne WAP, o dabartiniams mobiliesiems įrenginiams, bei jų ribotoms galimybėms (kurios vis gerėja),
- nemažą dalį mobiliųjų įrenginių yra sunku naudoti, tačiau tai galima laikyti laikina problema.

Šiuo metu WAP yra pati prieinamiausia, ir labiausiai paplitusi technologija mobiliuosiuose belaidžiuose įrenginiuose. Todėl yra tikslinga ją naudoti įvairiems elektroninės komercijos sprendimams, tiek jau egzistuojantiems, tiek ir naujai kuriamiems.

Šiame darbe daugiausiai susidursime su WAE (ang. *Wireless Application Environment*) [8], o ypač WML (ang. *Wireless Markup Language*) – XML pagrįsta apibendrinta žymų kalba, optimizuota ribotų galimybių tinklams bei įrenginiams. WML puslapis-dokumentas susideda iš dėklo (angl. *deck*) kuriame gali būti viena arba daugiau kortelių (angl. *cards*). Vartotojui iš karto atsiunčiamas visas dėklas su kortelių rinkiniu. Vienu metu vartotojas mato tik vieną iš dėkle esančių kortelių, tačiau paprastai gali patekti į kitas nuorodų, ar kitokių navigacijos priemonių pagalba. Kadangi įrenginiui atsiunčiamas visas dėklas (kortelių rinkinys) iš karto – perėjimui iš vienos kortelės į kitą nėra reikalingas papildomas kreipimasis į serverį (taip sutaupomas laikas kurį vartotojui reikėtų papildomai laukti). Tačiau jei navigacija vyksta ne tarp to pačio rinkinio kortelių – informacija siunčiama iš serverio.

3.1 Sprendimų taikymo WAP pagrindimas

Prieš pritaikant esamus elektroninės komercijos sprendimus WAP, arba prieš kuriant naujus elektroninės komercijos sprendimus reikėtų ištirti ar šiai elektroninės komercijos sričiai tinkama WAP technologija. Pateiksime keletą pavyzdinių parametrų, į kuriuos reikėtų atsižvelgti, norint panaudoti WAP technologiją elektroninės komercijos sprendimams:

- Ar sprendimas atitiks vartotojų poreikius? Vartotojai paprastai nėra linkę naudoti belaidžių interneto įrenginių, kai netoli yra prieinamas galingesnis įrenginys, suteikiantis greitesnę bei pilnesnę sprendimo prieigą per standartu jau tapusį žiniatinklį. Paprastai belaidžiai interneto įrenginiai yra naudojami tokiais atvejais kai keliaujama (tiek judėjimo metu, tiek ir tiesiog esant toli nuo įprastų namų), atostogaujama, pietaujama, ilsimasi, ir pan.
- Ar sprendimas yra skirtas vidiniam (ne viešai prieinamam) naudojimui? Tokie WAP taikymai gali būti vieni iš sėkmingiausių, nes leidžia darbuotojams prieiti prie vidinių įmonės programų be poreikio sustoti ir įsijungti nešiojamąjį kompiuterį. Dažnai daug

prasmingiau yra darbuotojui suteikti WAP įrenginį nei nešiojamąjį kompiuterį – sutaupomos lėšos, o darbuotojui nereikia nešioti didelio bei nepatogaus įrenginio.

- Ar yra naujų taikymų, sprendimų ar funkcionalumo, kuris įmanomas tik WAP technologijų pagalba? Taikymai naudojantys baigtis-baigtis tipo susijungimus bei pramogų sistemos gali sulaukti ypatingo pasisekimo, labiausiai dėl vietos ir laiko, kada yra naudojami bevieliai interneto įrenginiai.

Iš pavyzdinių parametrų, į kuriuos reikėtų atsižvelgti, norint panaudoti WAP technologiją elektroninės komercijos sprendimams matome, kad tiek prieš pritaikant esamus elektroninės komercijos sprendimus WAP, tiek ir prieš kuriant naujus elektroninės komercijos sprendimus reikėtų labai gerai ištirti ar šiai elektroninės komercijos sričiai iš principo tikslinga panaudoti WAP technologiją, o jei tikslinga – koks yra šio taikymo tikslas, į kokią vartotojų sritį yra taikoma, koks turės būti funkcionalumas ar jo apribojimai, bei daug kitų taikymo parametrų.

3.2 Sprendimų taikymo WAP kokybės kriterijai

Kaip ir daugumoje elektroninės komercijos sprendimų terpių, WAP atveju yra svarbu sukurti ir tiekti kokybiškus bei visapusiškai gerus sprendimus. WAP elektroninės komercijos sprendimams galima taikyti sekančius pavyzdinius kokybės vertinimo kriterijus:

- Sprendimai (ypač viešai prieinami) turi būti arba greitai veikiantys, arba pritraukiantys vartotoją. Nemaža dalis vartotojų moka už naudojimosi WAP laiką, todėl prisijungimas prie sprendimo bei atsijungimas iš jo turi būti trumpi. Jei to nesilaikoma, sprendimas turi būti kuo labiau pritraukiantis vartotoją – priešingu atveju sprendimas bus nenaudojamas bei neatneš naudos.
- Sprendimai turi nesuteikti vartotojui pernelyg daug pasirinkimų, labiausiai – dėl įrenginių apribojimų. Reikia numatyti labiausiai tikėtinus vartotojo pasirinkimus, bei padaryti juos lengvai pasiekiamus.
- Sprendimus reikia kurti taip, kad vartotojui reikėtų įvesti kuo mažiau duomenų, nes duomenų (pavyzdžiui tokių kaip tekstas) įvedimas daugumoje įrenginių yra labai nepatogus. Pavyzdžiui naudodami mobilųjį įrenginį vartotojai mielai tikrins esamus valiutų kursus, tačiau tikimybė jog vartotojas norės tuo pačiu ir atlikti valiutos keitimo operaciją yra nedidelė.
- Neretai sprendimai turėtų būti pagrįsti veiksmais, ir teikti informaciją kuri yra laikina ar kintanti. Dauguma vartotojų nebus linkę skaityti dokumentacijos, įvairių aprašymų ar knygų naudodami mobilųjį įrenginį, pvz. mobilųjį telefoną. Paprastai vartotojams reikės informacijos kuri yra jiems svarbi bei nauja, nesvarbu ar tai bus finansiniai pranešimai, ar

sporto varžybų rezultatai, ar artimiausia orų prognozė, ar neseniai gautos elektroninio pašto žinutės, ar tiesiog pasikeitimai kalendoriuje.

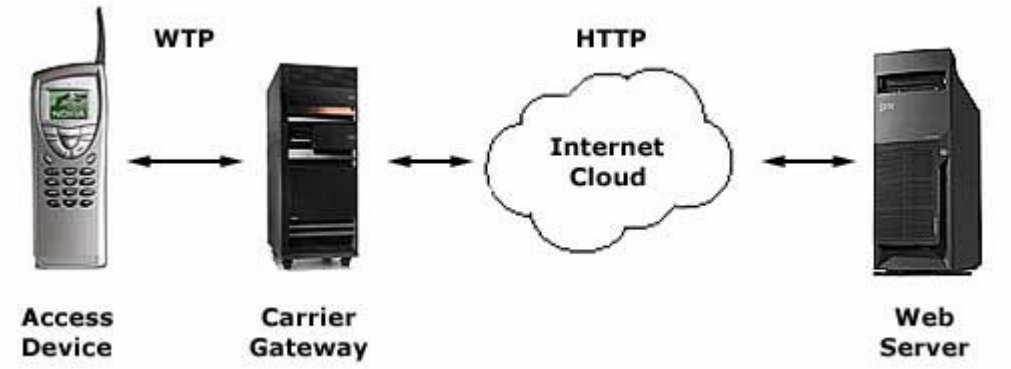
- Sprendimai turi būti lengvai naudojami – jei pasinaudot sprendimu bus žymiai sunkiau nei paskambinti, dauguma vartotojų gali tiesiog atsisakyti šio sprendimo naudojimo.

Apibendrinant reikia pastebėti, kad jei elektroninės komercijos sprendimai yra orientuoti į galutinį vartotoją, jie turi būti patrauklūs. Kadangi dauguma elektroninės komercijos sprendimų konkuruoja su analogiškas paslaugas teikiančiais sprendimais, reikia juos padaryti tiek kokybiškus ir gerus, kiek tik įmanoma. Todėl nors elektroninės komercijos sprendimuose dažnai gali ir nepavykti pritaikyti visų čia aprašytų (arba kitų sprendimo kokybę įtakojančių) parametrų, reikia stengtis į juos kuo labiau atsižvengti, nes sprendimo kokybė labai stipriai įtakoja ir jo sėkmę.

3.3 Sprendimų taikymo WAP specifika

Norint pritaikyti esamus elektroninės komercijos sprendimus WAP aplinkai, būtinas geras bazinių WAP aplinkos elementų supratimas. Programinės įrangos kūrėjus labiausiai dominti turėtų WAE (angl. *Wireless Application Environment*), įskaitant WML (angl. *Wireless Markup Language*), WML scenarijus (angl. *WML Script*), ir WTA (angl. *Wireless Telephone Application*).

Programinės įrangos kūrėjus nuo darbo su žemesniaisiais PĮ kūrimo sluoksniais „apsaugo“ WAP vartai Šie tarpiniai serveriai vadinami WAP vartais (angl. *gateway*) išverčia klientų užklausas iš belaidžio nešėjo protokolo į įprastines žiniatinklio užklausas.



3.2 pav. Įprastinė WAP panaudojimo schema, kurioje matoma tarpinio serverio pozicija

WML dažnai yra vadinamas HTML analogu WAP aplinkoje. Ir nors WML iš tikro yra skirtas ir WAP programų atvaizdavimui, jis taip pat turi savyje ir svarbių elementų kurie yra svetimi HTML, bei daugumai žiniatinklio programinės įrangos kūrėjų. Vieni iš labiausiai pastebimų elementų kurie yra nebūdingi HTML yra kintamieji, būsenos, įvykiai, o taip pat ir duomenų įvedimas, jo apribojimas.

WML yra XML pagrįsta kalba. Ji naudoja kortelių dėklo modelį informacijai vaizduoti. Tai leidžia WML dėklui (bylai) turėti savyje keletą skirtingų kortelių (puslapių). Tai puikiai tinka mobiliesiems įrenginiams, nes vienu duomenų perdavimu gaunami keli programos puslapiai. Sekančioje lentelėje palyginimui pateikiame paprastą WML dokumentą bei jo HTML atitikmenį.

3.1 lentelė Minimalių WML ir HTML dokumentų pavyzdžiai

WML	HTML
<?xml version="1.0"?> <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml"> <wml> <card id="card" title="Card 1"> Hello World! </card> </wml>	<html> <head> <title> Page 1 </title> </head> <body> Hello World! </body> </html>

Informacijos pateikimas naudojant WML nėra sudėtingas, tačiau labai svarbu jį atlikti teisingai, nes priešingai nei HTML atveju – menkiausia klaida reiškia jog WML puslapis bus iš viso nematomas, t.y. mobilieji įrenginiai negalės atvaizduoti net jo dalies.

Su WML dažnai naudojami ir WML scenarijai (angl. *WML Script*) bei WTA. WML scenarijai yra scenarijų kalba, kuri yra skirta atlikti tas pačias funkcijas, kurias HTML ir žiniatinklio naršyklių atveju atlieka JavaScript. Prieš naudojant WML scenarijus reikia įvertinti sekančius dalykus:

- dauguma, tačiau ne visi įrenginiai palaiko WML scenarijus
- jei vartotojų įrenginiai yra žinomi ir nekintantys (pvz. programa yra skirta įmonės vidiniam naudojimui) WML scenarijai gali būti labai naudinga technologija

WTA (angl. *Wireless Telephony Application*) sąsaja teikia taikymų karkasą telefonijos paslaugoms. Pavyzdžiui WTA leidžia WAP programoms turėti nuorodas, kurios gali inicijuoti skambučius, ir pan. Tiesa, kaip ir WML scenarijų atveju, ne visi įrenginiai palaiko šią technologiją, todėl daugumoje atveju pagrindinė naudojama technologija yra WML.

3.4 Infrastruktūros pakeitimai reikalingi WAP taikymui

Kadangi WAP vartai (tarpiniai serveriai kurie išverčia klientų užklausas iš belaidžio nešėjo protokolo į įprastines žiniatinklio užklausas) atlieka viską ko reikia kad mobilieji įrenginiai galėtų prieiti prie interneto resursų, reikalingi infrastruktūros pakeitimai yra minimalūs. Egzistuojantys žiniatinklio serveriai gali būti nesunkiai panaudojami ir WML prieigai teikti.

Paprastai mobiliųjų belaidžių įrenginių prieigai sukuriamas atskiras srities vidinis vardas, pvz:

- wap.srities_vardas.lt
- mweb.srities_vardas.lt
- wml.srities_vardas.lt

Šiuo metu labiausiai paplitę wap.srities_vardas.lt tipo pavadinimai.

Tačiau yra ir kita išeitis – mobilieji įrenginiai jungiasi įprastiniu adresu, tuo pačiu kurio jungiasi ir žiniatinklio naršyklės bei kitos programos. Tokiu atveju vartotojas nukreipiamas į atitinkamą turinį pagal jo naršyklės galimybes. Kas labiau tinka vartotojui – WML ar HTML galima spręsti iš jo naršyklės siunčiamų HTTP_ACCEPT ar HTTP_USER_AGENT. Tačiau jei svetainės apkrovimas yra labai didelis, toks sprendimas nerekomenduojamas – šis naršyklės galimybių tikrinimas bei vartotojų nukreipimas prie atitinkamo turinio reikalauja papildomų serverio resursų.

Norint panaudoti esamą žiniatinklio serverį WML prieigai teikti, paprastai reikia atlikti papildomą jo nustatymą. Jeigu serveryje dar nėra nustatyti MIME tipai skirti WAP – tai reikia padaryti. Žemiau lentelėje yra reikalingų MIME tipų sąrašas.

3.2 lentelė WAP naudojami MIME tipai

Failo plėtinys	MIME tipas
.wml	text/vnd.wap.wml
.wmlc	application/vnd.wap.wmlc
.wmls	text/vnd.wap.wmlscript
.wmlsc	application/vnd.wap.wmlscriptc
.wbmp	image/vnd.wap.wbmp

Labiausiai paplitusiame žiniatinklio serveryje „*Apache*“ tai daroma gan paprastai – nustatymų failo httpd.conf dalyje AddType papildomai įdedamos šios eilutės:

```
AddType text/vnd.wap.wml .wml

AddType image/vnd.wap.wbmp .wbmp

AddType application/vnd.wap.wmlc .wmlc

AddType text/vnd.wap.wmlscript .wmls

AddType application/vnd.wap.wmlscriptc .wmlsc
```

3.5 Sprendimų taikymo WAP architektūra

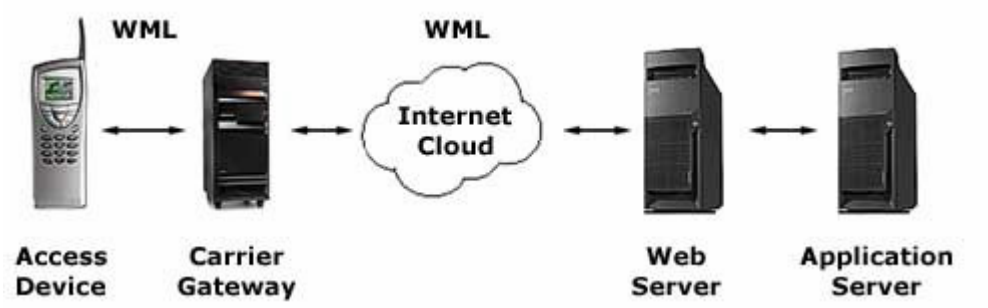
Norint sprendimui parinkti geriausiai poreikius atitinkančią architektūrą, reikia įvertinti tiek ir pačius sprendimo poreikius, tiek ir galimų architektūrų privalumus bei trūkumus. Belaidis internetas dažnai sukuria poreikį teikti panašius ar net tuos pačius duomenis keliais skirtingais pateikimo formatais (pvz. WML belaidžiams įrenginiams, bei HTML stacionariesiems).

Siūlome pasirinkti viena iš šių trijų galimų sprendimų taikymo WAP architektūrų:

- naudoti egzistuojančią architektūrą,
- naudoti XML tam kad atskirti programų atvaizdavimo (prezentacijos) sluoksnį,
- naudoti kodavimo serverį kuris bus atsakingas už tinkamą atvaizdavimą, t.y. dalis atvaizdavimo sluoksnio funkcijų paliekama kodavimo serveriui

3.5.1 Tipinė architektūra

Egzistuojančios architektūros panaudojimo schema atrodo taip:



3.3 pav. Egzistuojančios architektūros panaudojimas WAP prieigai

Žemiau pateiksime šios architektūros privalumus ir trūkumus.

Privalumai:

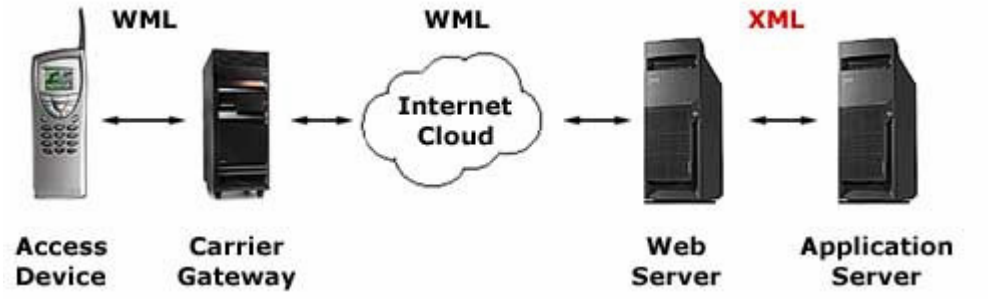
- esami sprendimai gali būti modifikuojami taip, kad atvaizduoti dalį informacijos pateikiamos žiniatinklyje, tačiau jau WML forma;
- gali būti kuriami nauji, optimizuoti belaidei aplinkai sprendimai, kurie naudoja esamą informaciją, esamą programinę įrangą bei aplinką;
- tai gali būti padaryta itin greitai, nenaudojant papildomų technologijų ar priemonių.

Trūkumai:

- sprendimai turi būti pakeisti kad vietoje HTML atvaizdavimą atliktų naudodami WML;
- programinės įrangos kūrėjai turi rūpintis naršyklės atpažinimu ir teisingo turinio formos panaudojimu;
- gali būti sudėtinga pritaikyti turinį tam tikriems įrenginiams.

3.5.2 Architektūra su išskirtu atvaizdavimo sluoksniu

Architektūros su išskirtu atvaizdavimo sluoksniu panaudojimo schema atrodo taip:



3.4 pav. Architektūros su išskirtu atvaizdavimo sluoksniu panaudojimas WAP prieigai

Iš vartotojo pusės ši architektūra gali atrodyti tokia pati. Tačiau žiniatinklio serveris duomenis iš programos serverio gauna XML formatu, ir gali priklausomai nuo užklausos ir kliento transformuoti šiuos XML duomenis į WML arba HTML dokumentus. Svarbu pastebėti jog žiniatinklio serveris ir

programos serveris nebūtinai turi būti skirtinguose serveriuose, t.y. tai gali būti vienas ir tas pats kompiuteris (todėl šis sprendimas nereikalauja papildomų išlaidų).

Privalumai:

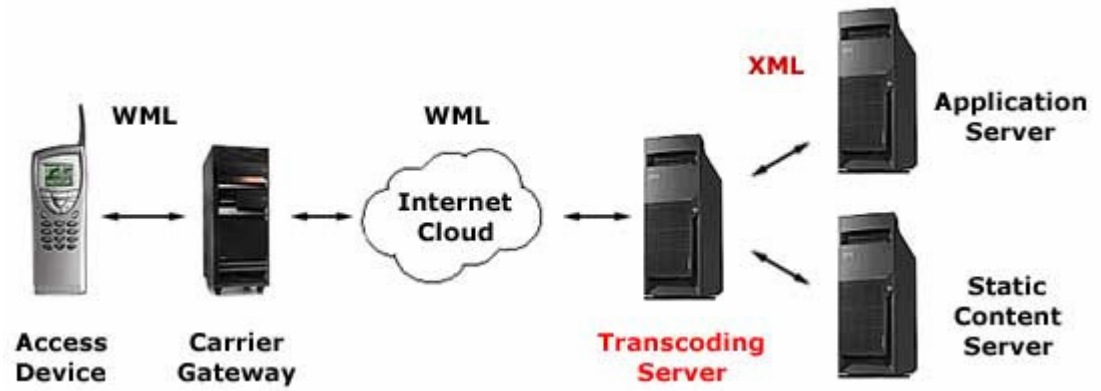
- patraukli architektūra, aiškiai atskirianti atvaizdavimą nuo programos logikos;
- naudoja populiarius ir atvirus formatus tam, kad transformuoti duomenis iš XML formato į atvaizdą pritaikytą kiekvienam įrenginiui;
- puikios galimybės transformuoti turinį į įvairius formatus, ne tik HTML ar WML;
- atvaizdavimo ir programos logikos atskyrimas leidžia atvaizdavimo pakeitimus atlikti lengviau ir greičiau.

Trūkumai:

- sudėtingesnė programinė įranga paprastai reiškia lėtesnį veikimą, ir didesnę sutrikimų tikimybę;
- koncepcinis XML elegantiškumas praktikoje gali ne visada pasiteisinti, nes geras WML atvaizdavimas nėra HTML atvaizdavimo poaibis; tai gali sukurti situaciją, kai XML failas yra naudojamas saugoti keletui skirtingų duomenų aibių, arba kai reikalingos itin sudėtingos XSL transformacijos. WML atvaizdavimo optimizavimas gali įtakoti tokių WML savybių panaudojimą, kurioms HTML tiesiog neturi analogų.
- atvaizdavimo sluoksnio kūrimas ir naudojimas reiškia darbą su 3 technologijomis – XML, XSL ir WML, vietoje to kad būtų dirbama vien tik su WML.
- didelė belaidžių įrenginių dalis turi labai ribotą atmintinės kiekį, o XSL nėra priemonių reguliuoti transformacijos imties dydį pagal sugeneruoto rezultato dydį.
- gerų WML programų atveju, riba tarp atvaizdavimo ir programos logikos yra dar mažesnė nei HTML atveju.

3.5.3 Architektūra su kodavimo serveriu

Architektūros su kodavimo serveriu panaudojimo schema atrodo taip:



3.5 pav. Architektūros su kodavimo serveriu panaudojimas WAP prieigai

Privalumai:

- kodavimo serveriai gali „protingai“ transformuoti turinį kiekvienam įrenginiui, pavyzdžiui grafiniai vaizdai gali būti automatiškai paverčiami nespalvotais mobiliesiems telefonams, o delniniams kompiuteriams paliekami spalvoti.
- kodavimo serverio įsigijimas programinės įrangos kūrėjams leidžia nekurti dalies funkcionalumo, tokio kaip vartotojo atpažinimas ar su vartotojo įrenginiu susijusi logika, ir patikėti šias funkcijas atlikti kodavimo serveriui;
- sprendimas labiau tinka didesnėms įmonėms, kurioms šis sprendimas nėra pernelyg brangus;

Trūkumai:

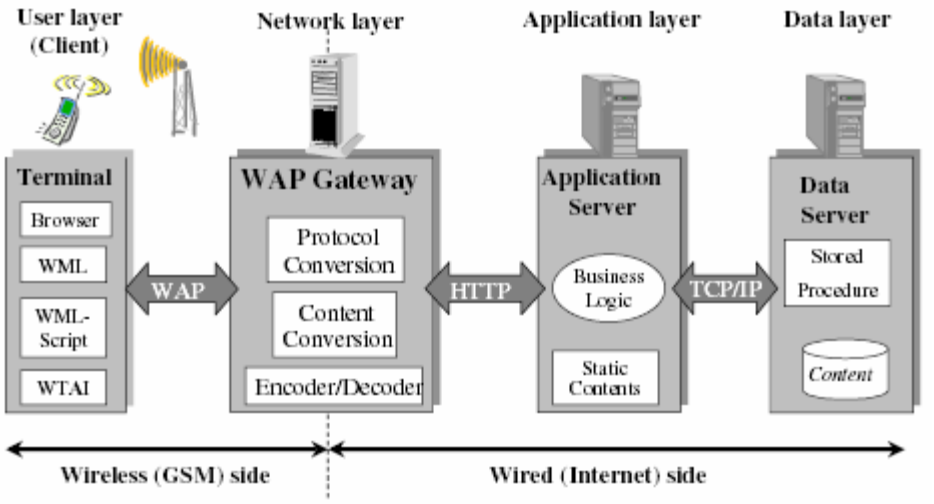
- kodavimo serveriai turi daugumą tų pačių trūkumų kaip ir anksčiau aptarta architektūra, naudojanti XML;
- kodavimo serveriai nėra išbaigtas belaidis sprendimas – dabartinio sprendimo automatinis kodavimas į WML daugumoje atvejų sugeneruos WML, kuris bus netinkamas jokiam belaidžiui įrenginiui.

3.5.4 Architektų sluoksninė analizė

Visas ankstesniuose skyriuose aprašytas sprendimų taikymo WAP architektūras sieja bendras tikslas – išplėsti paprastas WAP karkaso galimybes sudėtingesnėmis paslaugomis, tokiomis kaip transakcijos, būsenų valdymas po kliento atsijungimo ir pan. Apibendrinus visas ankstesniuose skyriuose aprašytas sprendimų taikymo WAP architektūras, jas galima įsivaizduoti kaip 4 sluoksnių sprendimą, kuris išnaudoja įprastus WAP bei žiniatinkliui skirtus komponentus, ir leidžia įgyvendinti šias užduotis:

- baigtis-baigtis (angl. *end-to-end*) tipo komunikavimą tarp mobiliojo terminalo (mikronaršyklės) ir programų serverio (t.y. programos logikos);
- nuotolinę prieigą prie duomenų bazių;
- transakcijų palaikymas, išnaudojant ir DBVS teikiamas transakcijų paslaugas;
- optimizuotą terminalo atsijungimo valdymą.

Sekančioje schemoje pavaizduotas apibendrintas 4 sluoksnių sprendimas.



3.6 pav. Apibendrinta 4 sluoksnių sprendimo schema

Keturi naudojami sluoksniai yra šie:

- Vartotojo (angl. *user*) sluoksnis – belaidis terminalas su WML mikro naršykle;
- Tinklo (angl. *network*) sluoksnis – atstovaujamas WAP vartų, veikiančių kaip HTTP tarpinis serveris;
- Programos (angl. *application*) sluoksnis – programų serveris, ir jame esanti logika;
- Duomenų (angl. *data*) sluoksnis – susideda iš DBVS serverių ir juose esančių duomenų bazių, o taip pat duomenų iš senesnių programų bei iš dalies struktūrizuotų duomenų (HTML, XML).

Sekančioje lentelėje pateikti sąsajų protokolai, kuriais vyksta komunikavimas tarp šių keturių lygių:

3.3 lentelė Sąsajų tarpusavio komunikavimo protokolai

Sąsaja tarp sluoksnių	Protokolas	Sritis
Vartotojo – Tinklo	WAP	Belaidis tinklas (GSM)
Tinklo – Programos	HTTP	Laidinis tinklas (Internetas)
Programos – Duomenų	TCP / IP	Laidinis tinklas (vietinis tinklas, LAN)

3.6 Sprendimų taikymo WAP atvejai

Galima nesunkiai išskirti šiuos du pagrindinius elektroninės komercijos sprendimų taikymo WAP atvejus:

- esami sprendimai,
- nauji sprendimai.

Tikslinga kiekvieną iš šių atvejų panagrinėti atskirai, o vėliau papildomai pateikti informaciją, tinkančią abiem atvejams.

3.6.1 Esami sprendimai

Esamų elektroninės komercijos sprendimų atveju atliekamas sistemos papildymas, o dažnai ir esamos dalies modifikavimas. Tai priklauso nuo konkrečios sistemos savybių. Tačiau jei sistema sukurta korektiškai – bus naudojamas atskiras sluoksnis atvaizdavimui, todėl dauguma pakeitimų bus atliekami būtent jame.

Galima išskirti du pagrindinius esamo elektroninės komercijos sprendimo modifikavimo atvejus – kai sistema pagrįsta šablonais, ir kai šablonai yra nenaudojami. Kai sistemoje naudojami šablonai, norint sukurti WAP prieigą pastangų reikia mažiau. Geriausiu atveju užteks paversti HTML šablonus į WML (metodika kaip tai reikia padaryti yra aprašyta sekančiuose skyriuose), sukurti modulį kuris nustatys kuriuos šablonus (WAP ar HTML) reikia naudoti konkrečiai naršyklei ar vartotojui, o taip pat jeigu reikia – paversti esamus grafinius elementus į WBMP formatą.

Jei šablonai nėra naudojami, yra du sprendimo būdai – arba juos pradėti naudoti, arba WAP prieigai reikia tiesiog sukurti naują programos arba atvaizdavimo sluoksnio dalį. Nors pirmasis būdas (šablonų naudojimas) yra universalesnis, jis turi ir pagrindinį su universalumu susijusį trūkumą – konkrečiu atveju specializuotas sprendimas visada yra geresnis nei universalus. Naudojant vien tik šablonus, nebus pilnai išnaudojamos WML teikiamos galimybės, XML atveju gali prireikti itin sudėtingų transformacijų ir kt.

Taip pat reikia atsižvelgti ir į tai, jog pats esamo (HTML grįsto) sprendimo keitimas į šablonus naudojantį (neskaičiuojant WML prieigos ir šablonų kūrimo), gali pareikalauti daug daugiau darbo, nei paprastas WAP prieigos pridėjimas.

Yra ir trečias esamo elektroninės komercijos sprendimo modifikavimo būdas, tai yra architektūros su kodavimo serveriu panaudojimas. Tačiau šis sprendimas turi daug trūkumų ir yra labai retai naudojamas (žr. ankstesnį skyrių „Architektūros su kodavimo serveriu panaudojimas“), todėl čia aptariamas nebus.

Akivaizdu, jog didžioji dalis elektroninės komercijos sprendimo modifikavimo priklauso nuo konkretaus atvejo, ir todėl negalima nustatyti bendros ir detalios procedūros, kaip tai reikia atlikti.

3.6.2 Nauji sprendimai

Kuriant naujus elektroninės komercijos sprendimus reikia nuspręsti kokia bus naudojama sprendimo architektūra (žr. ankstesnį skyrių „Sprendimų taikymo WAP architektūra“). Sistemos architektūra turėtų būti tokia, kad atvaizdavimo sluoksnis būtų atskirtas nuo likusių – priešingu atveju arba turėsime labai nekorektiškai sukurtą sistemą, ir didelį programos teksto pasikartojimą. Taip pat sistemoje turėtu aiškiai būti atskirtos WAP bei HTML turinio dalys.

Vienas iš paplitusių sprendimų kaip atskirti atvaizdavimą nuo kitų programos dalių – šablonų naudojimas. Tokiu būdu norint sukurti tiek HTML tiek WAP prieigą, o vėliau ir ją modifikuoti,

pastangų reikia mažiau. Geriausiu atveju sukurtus HTML šablonus užteks paversti į WML (metodika kaip tai reikia padaryti yra aprašyta sekančiuose skyriuose), sukurti modulį kuris nustatys kuriuos šablonus (WAP ar HTML) reikia naudoti konkrečiai naršyklei ar vartotojui, bei nuspręsti ar atvaizdų formato keitimas bus savalaikis (angl. *on-the-fly*), ar serveryje laikysime atskirus paveiksliukus WML ir HTML atvejams.

Tačiau nors šablonų naudojimas yra universalesnis būdas atvaizdavimo terpės daugialypiškumui spręsti, jis turi ir pagrindinį su universalumu susijusį trūkumą – konkrečiu atveju specializuotas sprendimas visada yra geresnis nei universalus. Naudojant vien tik šablonus, nebus pilnai išnaudojamos WML teikiamos galimybės, XML XSLT atveju gali prireikti itin sudėtingų transformacijų ir kt. Todėl reikėtų apsispręsti ar svarbiausia yra lengvas sprendimo modifikavimas (tuo atveju reikėtų naudoti šablonus), ar galutinis vartotojas, ir jam suteikiamos elektroninės komercijos sprendimo prieigos kokybės parametrai.

Akivaizdu, jog nemaža dalis naujo elektroninės komercijos sprendimo kūrimo su daugialype prieiga priklauso nuo konkretaus atvejo, ir todėl negalima nustatyti bendros ir detalios procedūros, kaip tai reikia atlikti.

3.7 HTML vertimas į WML

WML yra XML pagrįsta žymų kalba, kurios paskirtis yra apibrėžti turinio atvaizdavimą bei vartotojo sąsają įrenginiuose su gan ribotomis galimybėmis, pvz. mobiliuosiuose telefonuose. WML yra viena iš WAP sudedamųjų dalių.

Norint esamą sprendimą papildyti, sukuriant jam ir WAP prieigą, būtina šio proceso dalis yra HTML turinio vertimas į WML. Yra siūloma daug įvairios programinės įrangos kuri gali tai atlikti – tiek paversti atskirus statinius puslapius iš HTML į WML, tiek ir atlikti visa tai „nepastebimai“, universalių kodavimo serverių pagalba. Tačiau statinio HTML pavertimo i WML neužtenka jau vien todėl, kad internete yra labai mažai statinės informacijos, jau nekalbant apie elektroninės komercijos sprendimus, kuriuos sunku net įsivaizduoti statinių puslapių pavidalu. O didžiausi universalių kodavimo serverių trūkumai yra šie:

- universalių kodavimo serverių vertimo rezultatas iš HTML į WML paprastai yra visais aspektais žymiai prastesnis nei verčiant rankiniu būdu – tiek kodo teisingumo, tiek atvaizdavimo, tiek ir funkcionalumo;
- norint turėti universalių kodavimo serverį reikalinga brangi programinė (o neretai ir techninė) įranga.

Tiesa programinę įrangą, skirtą statinių puslapių vertimui iš HTML į WML, kartais įmanoma panaudoti pritaikant esamus elektroninės komercijos sprendimus WAP. Tai yra įmanoma tais atvejais, kai naudojami šablonai. Tačiau šios programinės įrangos, skirtos statinių puslapių vertimui iš HTML į

WML, panaudojimas priklauso nuo konkrečios naudojamos šablonų pritaikymo programinės įrangos, bei jos parametrų. Taip pat neretai būna ir taip, kad norint panaudoti programinę įrangą, skirtą statinių puslapių vertimui iš HTML į WML, reikia ją modifikuoti, kad keičiama būtų tik šablono HTML dalis, o dalis kuri nurodo vietą kintamai informacijai įrašyti turi likti nepakeista.

Taigi prieita išvada jog pritaikant esamus elektroninės komercijos sprendimus WAP, geriausia naudoti rankinį, arba iš dalies rankinį atvaizdavimo sluoksnio keitimą, keičiant terpę iš WWW į WAP, o jei konkrečiai – žymų kalbą iš HTML į WML. Tik taip bus pasiekiamas visais atžvilgiais kokybiškiausias rezultatas.

Todėl atliekant turinio žymų kalbos vertimą iš HTML į WML, būtina šiam procesui sudaryti metodiką bei taisykles, kurias ateityje bus galima tobulinti. Tačiau taisyklės turi būti apibendrintos, bei leidžiančios atlikti žymų kalbos vertimą iš HTML į WML tiek turint pilną ir teisingą HTML dokumentą, tiek ir tam tikrą dokumento fragmentą.

3.7.1 Vertimo metodika

Paprasčiausias ir lengviausias būdas kurį galima sugalvoti tam, kad paversti HTML į WML susideda iš dviejų ir vieno papildomo žingsnio. Pirmo žingsnio metu paimamas originalus HTML dokumentas, ir paverčiamas į DOM (angl. *Document Object Model*) atvaizdą, naudojant HTML nagrinėtoją (angl. *parser*). DOM yra medžio struktūra, atitinkanti visus žiniatinklio keliamus reikalavimus. Antrajame žingsnyje HTML DOM atvaizdas yra modifikuojamas, sukuriant DOM susidedantį iš WML elementų, pilnai laikantis visų WML standartų. Tai atliekama kiekvienam elementui. Pavyzdžiui HTML dokumentas prasideda *HTML* žyma (elementu-viršūne), kai tuo tarpu WML dokumentas prasideda elementu *WML*. Todėl antrajame žingsnyje, DOM struktūroje *HTML* elementas pakeičiamas *WML* elementu.

Dirbant su kiekvienu HTML elementu po vieną, juos galima suskirstyti į šias tris kategorijas:

- Elemento vertimas reikalauja unikalios logikos, pvz. HTML elementas ankstesniajame pavyzdyje.
- Elementas, ir visi „po juo“ esantys elementai su visu turiniu yra tiesiog ištrinami. Paprastai tai atliekama HTML elementams, kuriems WML tiesiog nėra analogų, arba tiems, kurių pernešimas į WML nėra prasmingas. Pavyzdžiui APPLET arba OBJECT HTML žymos į WAP nėra pernešamos, o yra tiesiog naikinamos.
- Elementas yra pakeičiamas elementais esančiais „po juo“. Tai reiškia jog turinys, esantis šiame elemente, yra paliekamas, tačiau pats elementas paliekamas. Pavyzdžiui SUP elementas, kuris nurodo, kad jame esantis tekstas turėtų būti atvaizduojamas mažomis raidėmis eilutės viršuje, t.y. „AAA^{BBB}“ reiškia sekantį atvaizdavimą – „AAABBB“. Toks atvaizdavimas WML yra negalimas, todėl „<SUP>tam tikras

tekstas</SUP>“ yra pakeičiamas tiesiog „tam tikras tekstas“. Žinoma išlieka ne tik viduje esantis turinys, bet ir kitos žymos, pvz. „^{1<B>2</B>3}“ bus pakeičiamas „1<B>2</B>3“.

Pirmos kategorijos elementų apdorojimui reikalinga speciali logika, kuri gali būti skirtinga kiekvienam elementui. Tačiau antros bei trečios kategorijos elementų apdorojimas gali būti atliekamas vieno apdorojimo modulio, kuriam pateikiamas tam tikrų apdorotinų HTML elementų sąrašas. Taigi moduliui, atsakingam už nereikalingų elementų šalinimą arba pakeitimą jų turiniu, reikėtų tik pateikti HTML elementų, kurių apdorojimas vyks tokiu būdu, sąrašą.

Kai HTML dokumentas yra paverstas į tikslo žymų kalbą (WML), gali būti atliekamas papildomas trečiasis vertimo žingsnis, kurį pavadinsime teksto iškarpymu. Yra bent keletas priežasčių, kodėl tai yra naudinga:

- Nors bendras tam tikro HTML dokumento vertimas į WML ir baigsis sintaksiškai teisingu WML dokumentu, rezultatas gali būti nepatogus naudojimui.
- Išverstas dokumentas gali savyje turėti didelį kiekį nereikalingos informacijos, kuri vartotojui reikalingą informaciją padaro sunkiai surandama ir prieinama. Pavyzdžiui galingos, įprastiems kompiuteriams skirtos naršyklės atvaizduotame puslapyje, yra prasminga ir įprasta turėti nuorodas į kiekvieną produktą ar įmonės padalinį jau pačiame puslapio viršuje. Tačiau mikro naršyklių vartotojams tai gali būti labai nepatogu, nes norint pasiekti reikalingą informaciją, kaskart reikia praeiti (angl. *scroll*) pro įvairias nuorodas, esančias puslapio viršuje.

Žemiau pateiksime tris būdus, kaip galima atlikti teksto iškarpymo žingsnį:

- Kadangi WML yra XML pagrįsta kalba, vartoti XML stilių lentelės,
- Naudoti įprastinius teksto apdorojimo metodus, tam kad ištrinti nereikalingą tekstą,
- Tai atlikti tiesiogiai DOM struktūroje.

Kadangi vien šablonų naudojimas elektroninės komercijos sprendimų prieigai WAP sukurti yra nerekomenduojamas (labiausiai dėl WAP įrenginių, bei WML kalbos specifikos), o pirmą ir paskutinį būdą panaudoti ne HTML ar šablonams yra neįmanoma (nes reikėtų keisti programos kodą, pvz. PHP), lieka tik trečiasis būdas.

3.7.2 Žymų vertimas

Manau jog HTML žymas verčiant į WML žymas, bendru atveju reikia žinoti:

- HTML žymos pavadinimą
- WML pavadinimą
- Žymų savybes kurias norėsime išsaugoti verčiant informaciją HTML į WML

- Savybių HTML žymose atitikmenis WML žymėse (t.y. ar reikia pernešant HTML žymos savybę į WML, pakeisti ir savybės vardą)
- Papildomas žymas atsirandančias vertimo metu (skirta tam atvejui, kai viena HTML žyma keičiama tam tikra WML žymų kombinacija, pvz. <H1>tekstas</H1> → <P><BIG>tekstas</BIG></P> ir pan.)
- Žymos nedalomumas (t.y. ar galima informaciją esančią šioje žymoje perkelti į sekančią WML kortelę, arba net dėklo bylą, ar ne)

Taip pat manau jog ši informacija gali būti naudinga vertimui atlikti (priklausomai nuo paskirties bei sudėtingumo):

- HTML žymos tipas
- Žymų, galimų tam tikroje žymoje, sąrašas.

HTML žymų dokumento, ar jo dalies, vertimui į WML siūlau sekančioje lentelėje esančias HTML žymų vertimo į WML taisykles.

3.4 lentelė HTML žymų vertimo į WML žymas taisyklių lentelė

HTML žyma	Žymos tipas	Veiks-mas	WML žyma	Išsau-gomos savybės	Savybės vardo keitimas	Vidinės žymos	Papil-domos žymos	Uni-kali	Neda-loma
html	Žymų kalbos	pakeisti	wml						
wml	Žymų kalbos	palikti				head, template, card		Taip	
head	Antraštės	palikti	meta, access					Taip	
template	Antraštės	palikti	do, onevent					Taip	
title	Antraštės	ištrinti							
base	Antraštės	pakeisti	-tuščia-						
style	Antraštės	ištrinti							
script	Antraštės	ištrinti							
body	Struktūros	pakeisti	card						
card	Struktūros	palikti				onevent, timer, do, p, pre		Taip	
h1	Struktūros	pakeisti	p				big,strong		Taip
h2	Struktūros	pakeisti	p				big		Taip
h3	Struktūros	pakeisti	p				strong		Taip
h4	Struktūros	pakeisti	p						Taip
h5	Struktūros	pakeisti	p						Taip
h6	Struktūros	pakeisti	p						Taip

HTML žyma	Žymos tipas	Veiks-mas	WML žyma	Išsau-gomos savybės	Savybės vardo keitimas	Vidinės žymos	Papil-domos žymos	Uni-kali	Neda-loma
li	Struktūros	pakeisti	p						
dt	Struktūros	pakeisti	p						
dd	Struktūros	pakeisti	p						
div	Struktūros	pakeisti	p						
p	Struktūros	palikti		align		em, strong, b, i, u, big, small, br, img, anchor, a, table, input, select, fieldset, do			
br	Struktūros	palikti				EMPTY			
pre	Struktūros	palikti				a, br, i, b, em, strong, input, select			
tt	Struktūros	pakeisti	pre						
table	Lentelių	palikti		title, align		tr			
caption	Lentelių	ištrinti							
tr	Lentelių	palikti				td			
th	Lentelių	pakeisti	td						
td	Lentelių	palikti				em, strong, b, i, u, big, small, br, img, a, anchor			
a	Nuorodų	palikti		id, name, href, title, accesske y	name -> id	br, img			
anchor	Nuorodų	palikti		id, title, accesske y		br, go, img			
img	Nuorodų	palikti		id, src, alt, align		-tuščia-			
frame	Nuorodų	pakeisti	p						
area	Nuorodų	pakeisti	p						
em	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			

HTML žyma	Žymos tipas	Veiks-mas	WML žyma	Išsau-gomos savybės	Savybės vardo keitimas	Vidinės žymos	Papil-domos žymos	Uni-kali	Neda-loma
strong	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			
b	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			
i	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			
u	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			
big	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			
small	Stiliaus	palikti				em, strong, b, i, u, big, small, br, img, anchor, a, table			
do	Įvykių	palikti		type, label, name, optional		go, prev, refresh, noop			
onevent	Įvykių	palikti		type		go, prev, refresh, noop			
go	Užduočių	palikti		href, method, enctype, sendrefe rer, cache-control, accept-charset		postfield, setvar			
postfield	Užduočių	palikti		name,		-tuščia-			

HTML žyma	Žymos tipas	Veiks-mas	WML žyma	Išsau-gomos savybės	Savybės vardo keitimas	Vidinės žymos	Papil-domos žymos	Uni-kali	Neda-loma
				value					
setvar	Užduočių	palikti		name, value		-tuščia-			
prev	Užduočių	palikti				setvar			
refresh	Užduočių	palikti				setvar			
noop	Užduočių	palikti				-tuščia-			
form	Sąsajos	pakeisti	-tuščia-						
select	Sąsajos	palikti		title, name, value, multiple		optgroup, option			
optgroup	Sąsajos	palikti		title		optgroup, option			
option	Sąsajos	palikti		title, value		onevent,			
input	Sąsajos	palikti		name, type, value, title, size, maxleng th		-tuščia-			
fieldset	Sąsajos	palikti		title		em, strong, b, i, u, big, small, br, img, anchor, a, table, input, select, fieldset, do			
timer	Sąsajos	palikti		name, value		-tuščia-			

HTML žymas nepaminėtas šioje lentelėje siūloma ištrinti (su visu viduje esančiu turiniu), arba pakeisti trumpu tekstu, pažyminčiu kas buvo ištrinta.

Verčiant žymas iš HTML į WML (priklausomai nuo paskirties bei sudėtingumo) taip pat gali prireikti sužinoti įprastinę bei norimą žymę – tėvą, t.y. žinoti kokia WML žyma turėtų būti kokios žymos viduje. Ši informacija pateikiama sekančioje lentelėje:

3.5 lentelė WML elementų aukštesniųjų elementų lentelė.

WML žyma	Žymos tipas	Aukštesnė WML žyma
wml	Antraštės	-nėra-
head	Antraštės	wml
meta	Antraštės	head
access	Antraštės	head
template	Antraštės	wml
onevent	Antraštės	template
card	Struktūros	wml
p	Struktūros	card
pre	Struktūros	card
br	Struktūros	p
table	Lentelių	p
tr	Lentelių	table
td	Lentelių	tr
a	Žymų	p
anchor	Žymų	p
img	Žymų	p
b	Stiliaus	p
i	Stiliaus	p
u	Stiliaus	p
strong	Stiliaus	p
em	Stiliaus	p
big	Stiliaus	p
small	Stiliaus	p
select	Sąsajos	p
option	Sąsajos	select
optgroup	Sąsajos	select
do	Sąsajos	p
input	Sąsajos	p
fieldset	Sąsajos	p

Šios taisyklės skirtos bendrajam žymų vertimo iš HTML į WML atvejui. Tačiau priklausomai nuo verčiamo turinio ir jo paskirties, o taip pat ir nuo norimo turinio modifikavimo laipsnio, šias taisyklės galima šiek tiek papildyti, ar pakeisti, ir žymas versti kiek kitaip nei aprašyta pradinėje taisyklių versijoje.

3.7.3 Grafinių elementų vertimas WAP aplinkai

WAP (angl. *Wireless Application Protocol*) protokolo paskirtis, kaip ir specifikacijos, stipriai skiriasi nuo WWW žiniatinklio. Visų skirtumų priežastis yra ta, kad turinys perduodamas belaidžiais tinklais, ir yra skirtas mobiliesiems įrenginiams. Tiek įvairūs belaidžiai tinklai, tiek ir mobilieji įrenginiai turi aibę apribojimų, kurių neturi dauguma kompiuterių su įprastine interneto prieiga. Vienas iš šių apribojimų – mobiliųjų įrenginių grafinio atvaizdavimo galimybės (tiek dėl įrenginių ekranų, tiek

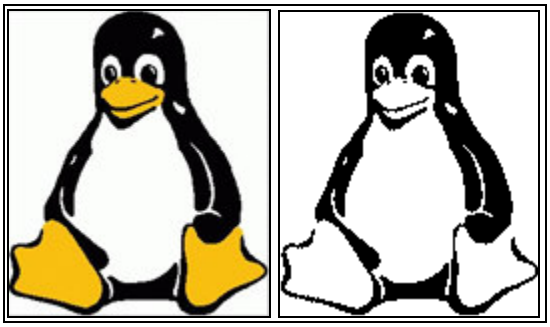
ir dėl skaičiavimų, reikalingų paveiksliukams atvaizduoti). Todėl WML turinio atveju yra naudojami WBMP formato grafiniai vaizdai.

WBMP (angl. *Wireless BitMap*) vaizdų formatas gerokai skiriasi nuo šio metu pasauliniame interneto žiniatinklyje paprastai naudojamų formatų. Šio skirtumo priežastis – ribotos mobiliųjų įrenginių grafinio atvaizdavimo galimybės, o taip pat ir šio atvaizdavimo kokybės poreikis.

WBMP atveju yra naudojama TypeField reikšmė, kuri aprašo kodavimo informaciją, tokią kaip taškų bei paletės organizavimą, animaciją bei suspaudimą. Taip pat naudojamas laukas ImageType, kuris nurodo paveikslo charakteristikas pagal WAP specifikaciją. Tačiau vienintelė WBMP įrenginių palaikoma ir specifikacijose nurodyta galima ImageType reikšmė šiuo metu yra 0, kas reiškia sekančius parametrus:

- nėra jokio suspaudimo,
- spalvai nurodyti naudojamas 1 bitas,
- gyliui nurodyti naudojamas 1 bitas.

Žemiau pateiksime keletą grafinių elementų vertimo WAP aplinkai pavyzdžių



3.7 pav. Paprasto grafinio elemento vertimas į WBMP

3.8 pav. Vertimas į WBMP – paprastas bei naudojant sklaidą (angl. *error diffusion*)

Paskutiniame pavyzdyje akivaizdžiai matome, jog net ir ribotos WBMP galimybės gali ne visada būti išnaudojamos: pvz. sklaidos (angl. *error diffusion*) naudojimas gali šiek tiek pagerinti kai kurių atvaizdų kokybę.

3.7.4 Vartotojo sąsajos elementų vertimas

Sekančioje lentelėje pateikti pavyzdžiai, kaip reikėtų versti dažnai pasitaikančius vartotojo sąsajos elementus iš HTML žymų kalbos į WML žymų kalbą.

3.6 lentelė Tipinių HTML sąsajos elementų vertimas į WML

HTML žymų kalbos fragmentas	WML žymų kalbos fragmentas
<form method="Get" action="A"> <input name="B"> <input type="submit" label="C"> </form>	<input name="B"/> <do type="accept" label="C"> <go href="A?B=$(B)"/> </do>
<form method="Post" action="A"> <input name="B"> <input type="submit" label="C"> </form>	<input name="B"> <do type="accept" label="C"> <go href="A" method="Post"> <postfield name="B" value="$(B)"/> </go> </do>
<Meta HTTP-equiv="refresh" content="A ; URL=B">	<onevent type="ontimer"> <go href="B"/> </onevent> <timer value="A"/>
<texarea name="A"> </textarea>	<input name="A"/>
<select name='A'> <option value='B'>C ... </select>	<select name='A'> <option value='B'>C</option> ... </select>

3.8 Kitos sprendimų taikymo WAP problemos

Vienas iš dalykų apie kuriuos būtina pagalvoti tiek kuriant naujus elektroninės komercijos sprendimus, tiek ir pritaikant elektroninės komercijos sprendimus WAP, yra sesijų valdymas. Belaidžių įrenginių atveju pagrįsti sesijų valdymą slapukais (angl. *cookies*) negalima, nes ne visi įrenginiai juos palaiko. Galimi sekantys šios problemos sprendimo variantai:

- URL „perrašymas“, t.y. reikia papildomai perdavinėti sesijos ID,
- naudoti „paslėptus“ (angl. *hidden*) laukus ,
- programinę įrangą suprojektuoti taip kad ji neturėtų būsenų (angl. *stateless*).

Jei būsenos yra būtinos tam tikram elektroninės komercijos sprendimui, tada reikia įdėmiai peržiūrėti ir įsitikinti jog sesijų valdymas yra tinkamas, ir tokiu būdu būsenos yra sėkmingai perduodamos.

Spartinimas (angl. *caching*) taip pat yra labai svarbus belaidžiams įrenginiams, nes:

- reikia spartinti tiek duomenų kiek tik įmanoma, tam kad išvengti ilgai trunkančių užklausų – atsakymų su serveriu, tuo pačiu minimizuojant laiką kurį vartotojui reikės laukti,
- belaidžiai įrenginiai gali lengvai būti pamesti arba pavogti – todėl labai svarbu užtikrinti jog slapti asmeniniai duomenys nėra spartinami, ir nelieka spartinančiojoje atmintinėje.

Puslapių spartinimui yra naudojami du pagrindiniai metodai – HTTP antraštės bei META žymos. HTTP antraštės turi būti atpažįstamos taip pat ir tokių įrenginių kaip tarpiniai serveriai (angl. *proxy*) ar tinklo spartintojai (angl. *network cache*). Nustatant HTTP Expires antraštę į tam tikrą ateities datą, reikš kad puslapiai bus spartinami iki nurodytos datos. Kita vertus, norint kad tam tikri puslapiai nebūtų spartinami, antraštėje reikia nurodyti jau praėjusią datą.

Puslapiams kurių reikia nespartinti, į antraštę taip pat reikia įrasyti:

```
Cache-Control: no-cache, must-revalidate

Pragma: no-cache
```

Taip pat galima nurodyti puslapio spartinimą ir naudojant vartotojo agento META žymas. META žymos yra atpažįstamos WAP įrenginių, bet ne kitų tinklo įrenginių (pvz. jau anksčiau minėti kaip tarpiniai serveriai ar tinklo spartintojai). Žemiau pateiktas pavyzdys, kaip naudojant META žymas nurodyti vartotojo įrenginiui jog puslapio nespartinti (t.y. jog puslapio „galiojimo laikas“ jau baigėsi) :

```
<meta forua="true" http-equiv="Cache-Control" content="max-age=0"/>
```

Spartinimo atveju reikia suprasti ir tai, jog WML spartinimas tiek WAP įrenginiuose, tiek ir tinklo įrenginiuose veikia dėklus, o ne atskiras korteles esančias dėkluose.

Svarbu ir tai, jog WML įrenginiuose nėra bendros vartotojo aplinkos (angl. *look and feel*), ir yra gausu neatitikimų, kas sukelia daug nepatogumų belaidės programinės įrangos kūrėjams. Kai kurie iš įrenginių palaiko WML pirmtaką HDML (angl. *Handheld Device Markup Language*), kai kurie – senesniąsias WML versijas, dar kituose WML atvaizdavimas yra tiesiog prastai įgyvendintas.

Dėl šių priežasčių reikėtų nuspręsti kokius įrenginius kuriamas sprendimas turės palaikyti, ir tada su tais įrenginiais atlikti bandymus. Nors yra daug įrenginių imitatorių, jie dažnai neatspindi visų įrenginių charakteristikų, todėl baigiamajam testavimui nereikėtų pasitikėti imitatoriais. Tokio testavimo rezultatai dažnai gali būti klaidingi, nes imitatoriai yra linkę būti labiau atsparūs ir tolerantiški klaidoms nei tikrieji įrenginiai.

3.9 Sprendimų taikymo WAP išvadų santrumpa

Tiek kuriant naujus elektroninės komercijos sprendimus, tiek ir pritaikant egzistuojančius elektroninės komercijos sprendimus WAP, reikia įvertinti sekančius dalykus:

- įsitikinti jog sprendimas yra pritaikomas WAP aplinkai, ir jog šis pritaikymas yra tikslingas;
- sprendimą suprojektuoti ir įgyvendinti reikia taip, kad jis būtų patogus, greitas ir patrauklus vartotojui;

- pasirinkti teisingą architektūrinį sprendimą: daugumoje atveju bus naudojama esama architektūra, tačiau jei reikalavimai ir galimybės yra didesni – reikia apsvarstyti XML arba kodavimo serveriais pagrįstus sprendimus;
- atkreipti dėmesį į naujas saugumo, spartinimo ir sesijų problemas, atsiradusias naudojant WAP;
- nuspręsti kokie įrenginiai bus palaikomi kuriamo sprendimo, ir tada juos naudojant atlikti bandymus (pradinius bandymus galima atlikti ir imitatorių pagalba, tačiau galutiniams būtina naudoti pačius įrenginius);
- prieš įdiegiant ir pateikiant naudojimui galutinį sprendimą, reikia įdiegti sprendimą į viešai prieinamus serverius ir atlikti vartotojiškus bandymus.

3.10 Vartotojo agento nustatymas

Paprastai mobiliųjų belaidžių įrenginių prieigai sukuriamas atskiras srities vidinis vardas, pvz wap.srities_vardas.lt. Tačiau dažnai naudojama ir kita išeitis – mobilieji įrenginiai jungiasi įprastiniu adresu, tuo pačiu kurio jungiasi ir žiniatinklio naršyklės bei kitos programos. Tokiu atveju vartotojas nukreipiamas į atitinkamą turinį pagal jo naršyklės (vartotojo agento) galimybes. Kas labiau tinka vartotojui – WML ar HTML galima spręsti iš jo naršyklės siunčiamų HTTP_ACCEPT ar HTTP_USER_AGENT.

Tačiau jei svetainės apkrovimas yra labai didelis, toks sprendimas nerekomenduojamas – šis naršyklės galimybių tikrinimas bei vartotojų nukreipimas prie atitinkamo turinio reikalauja papildomų serverio resursų.

Siūlau naudoti sekantį būdą automatiškai nustatyti pageidaujamo turinio tipą:

- **WML** - jeigu vartotojo naršyklės HTTP_ACCEPT (palaikomų duomenų tipų, priimamų HTTP protokolu sąrašas) aibėje yra tipas „text/vnd.wap.wml“ ir nėra tipo „text/html“
- **HTML** – kitais likusiais atvejais

Tačiau rekomenduoju papildomai sukurti adresus, kuriais galima tiesiogiai pasiekti norimą sistemos terpę, „apeinant“ automatinį turinio tipo nustatymą. Tai gali būti naudinga tais atvejais, kai naršyklė palaiko ir WML ir HTML, o vartotojas nori matyti WML versiją.

3.11 Prototipo kūrimas – WWW terpė

Kaip pažymėta įžangoje, šiame darbe bus bandoma sukurti elektroninės parduotuvės turinčios daugialypę prieigą prototipą, t.y. sprendimas turės prieigą tiek įprastiems ir delniniams kompiuteriams, tiek ir mobiliesiems telefonams.

Tačiau norint pademonstruoti esamo elektroninės komercijos sprendimo pritaikymą WAP terpei, iš pradžių bus sukurtas elektroninės parduotuvės prototipas vien WWW / HTML terpei, o vėliau

modifikuojamas į daugiaterpį, kurio viena iš terpių – WAP / WML. Šiame skyriuje apibendrintai apžvelgsime bendrą abiem terpėms informaciją.

Elektroninė parduotuvė internete skirta prekių bei paslaugų pardavimui elektroniniu būdu. Šiame darbe yra nagrinėjama kaip elektroninę parduotuvę būtų galima pasiekti, ir jos teikiamomis paslaugomis naudotis ir portatyviųjų įrenginių, tokių kaip delniniai kompiuteriai, mobilieji telefonai ir kt. pagalba.

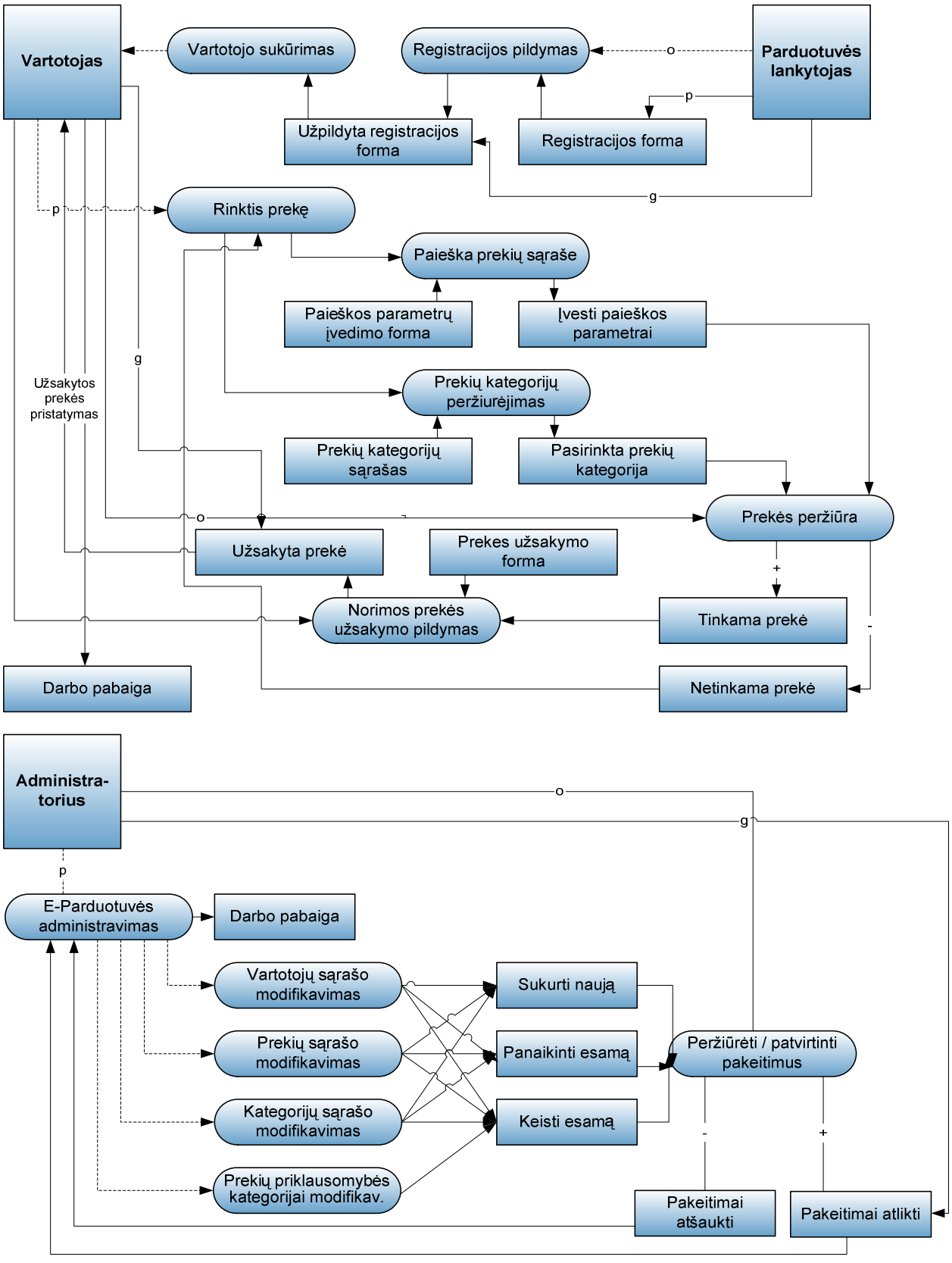
Elektroninė parduotuvė yra dažniausiai sutinkama elektroninės komercijos apraiška pasauliniame interneto tinkle. Sistemoje bus dviejų lygių vartotojai – paprastas vartotojas bei administratorius.

- Vartotojas – informacinėje sistemoje (prekės užsakymo metu) registruojamas asmuo, galintis peržiūrėti prekių sąrašą, prekių katalogų sąrašą, atlikti prekių paiešką, bei užsakyti prekes.
- Administratorius – tai informacinėje sistemoje užregistruotas asmuo, galintis modifikuoti prekių, prekių kategorijų, bei vartotojų sąrašus, t.y. keisti esamus, juos šalinti arba sukurti naujus. Taip pat paskirstyti prekes kategorijoms, keisti vartotojų teises.

Pagrindiniai panaudojimo atvejai:

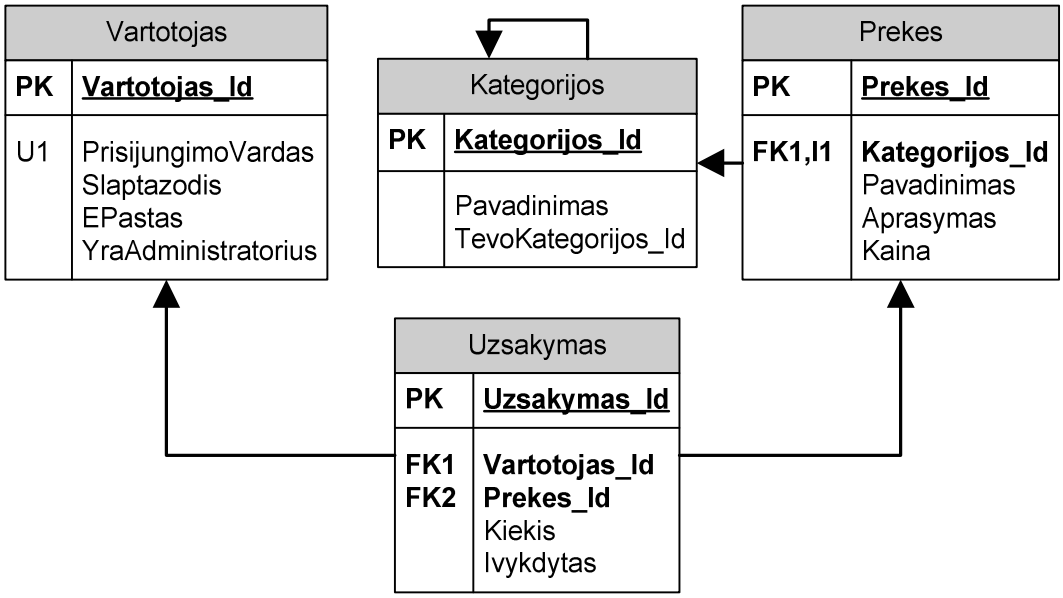
- Pasinaudoti el. parduotuve
  - Registruotis
  - Peržiūrėti prekių sąrašą
  - Užsisakyti norimą prekę
  - Ieškoti prekių sąraše
- Administruoti el. parduotuvę (gali tik administratorius)
  - Modifikuoti vartotojų sąrašą
  - Modifikuoti prekių sąrašą
  - Modifikuoti kategorijų sąrašą
  - Keisti prekės priklausomybę kategorijai

3.11.1 Organizacijos modelis



3.9 pav. Organizacijos modelis

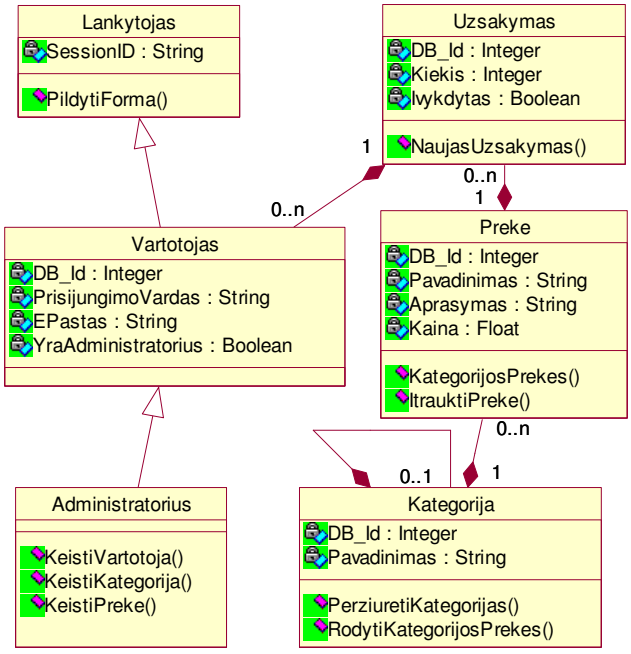
3.11.2 Esybių sąryšių diagrama



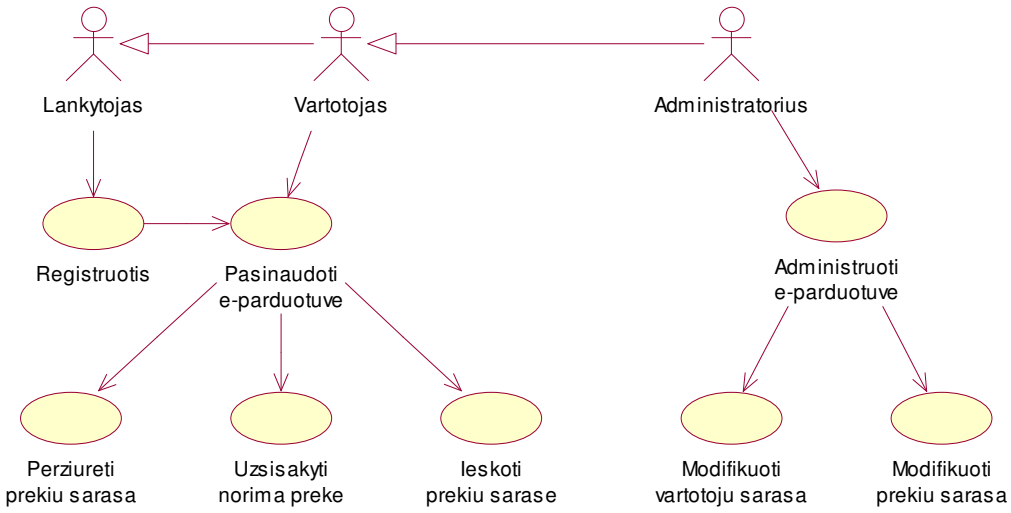
3.10 pav. Esybių sąryšių diagrama

3.11.3 Objektiniai modeliai

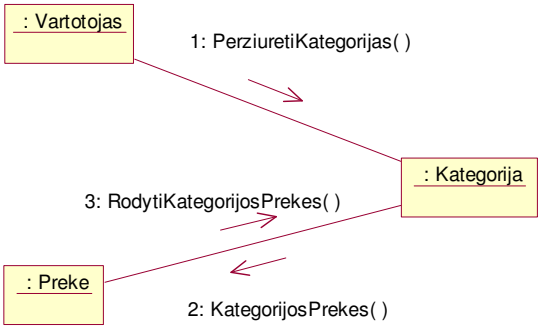
Šiame skyriuje pateikiami galimi elektroninės parduotuvės objektiniai modeliai.



3.11 pav. Klasių diagrama

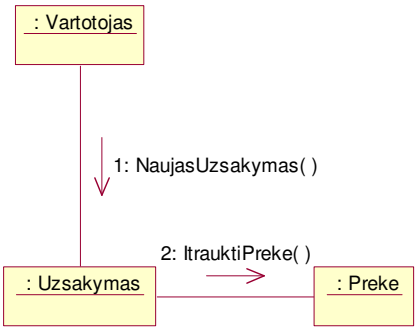


3.12 pav. Panaudojimo atvejų diagrama

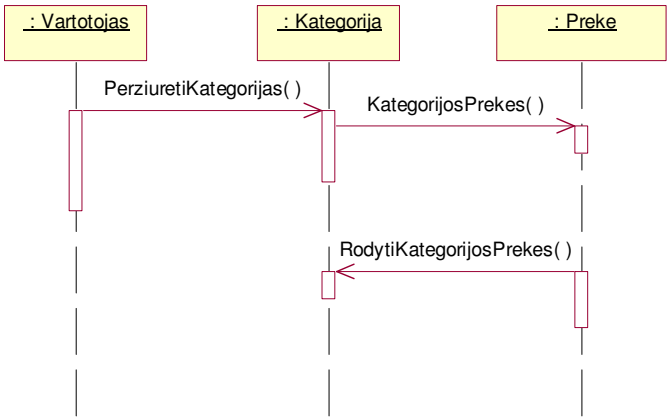


3.13 pav. Bendradarbiavimo diagramos – prekes pasirinkimo diagrama

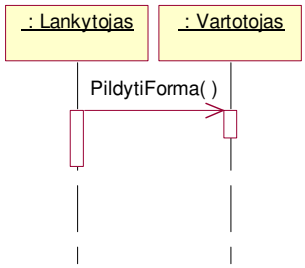
3.14 pav. Bendradarbiavimo diagramos – lankytojo registracijos diagrama



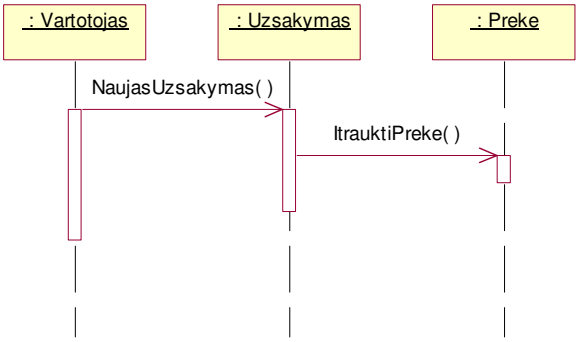
3.15 pav. Bendradarbiavimo diagramos – užsakymo diagrama



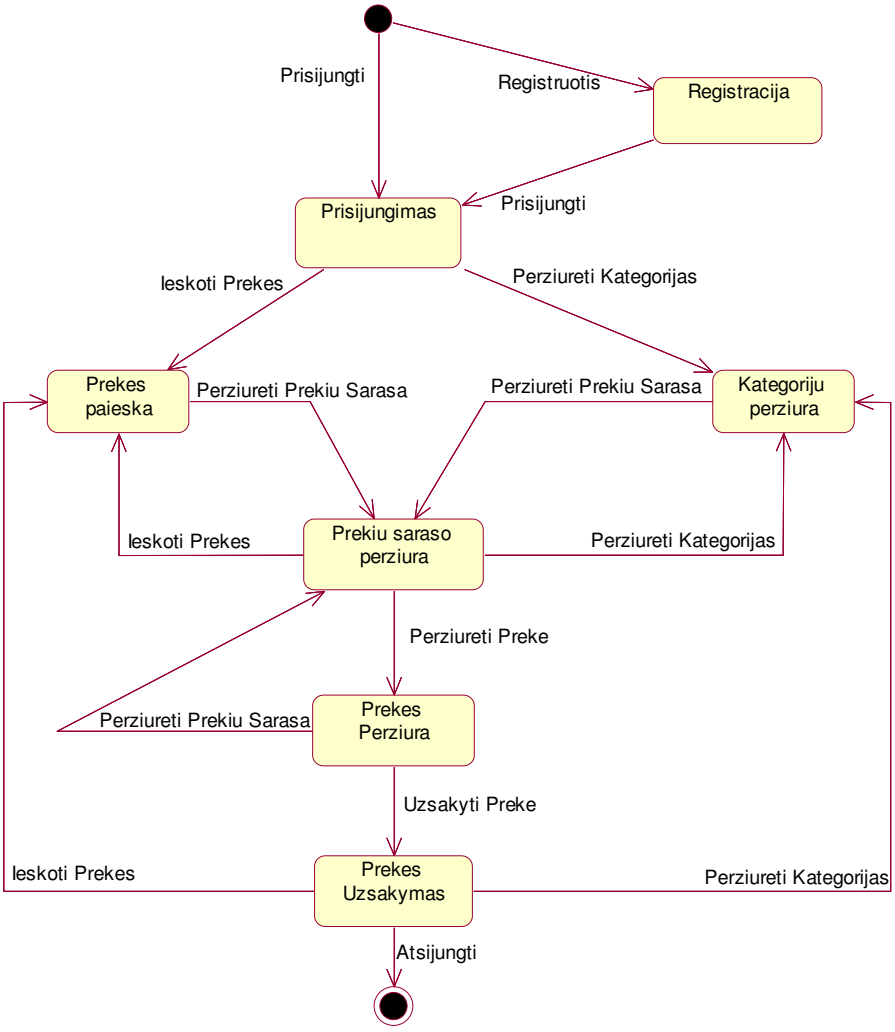
3.16 pav. Prekes pasirinkimo diagrama



3.17 pav. Lankytojo registracijos diagrama



3.18 pav. Užsakymo diagrama



3.19 pav. Būsenų diagrama

3.11.4 Kanoninė schema

Vartotojas

Vartotojas_Id	PrisijungimoVardas Slaptazodis EPastas YraAdministratorius

Uzsakymas

Uzsakymas_Id	Kiekis Ivykdytas

Prekes

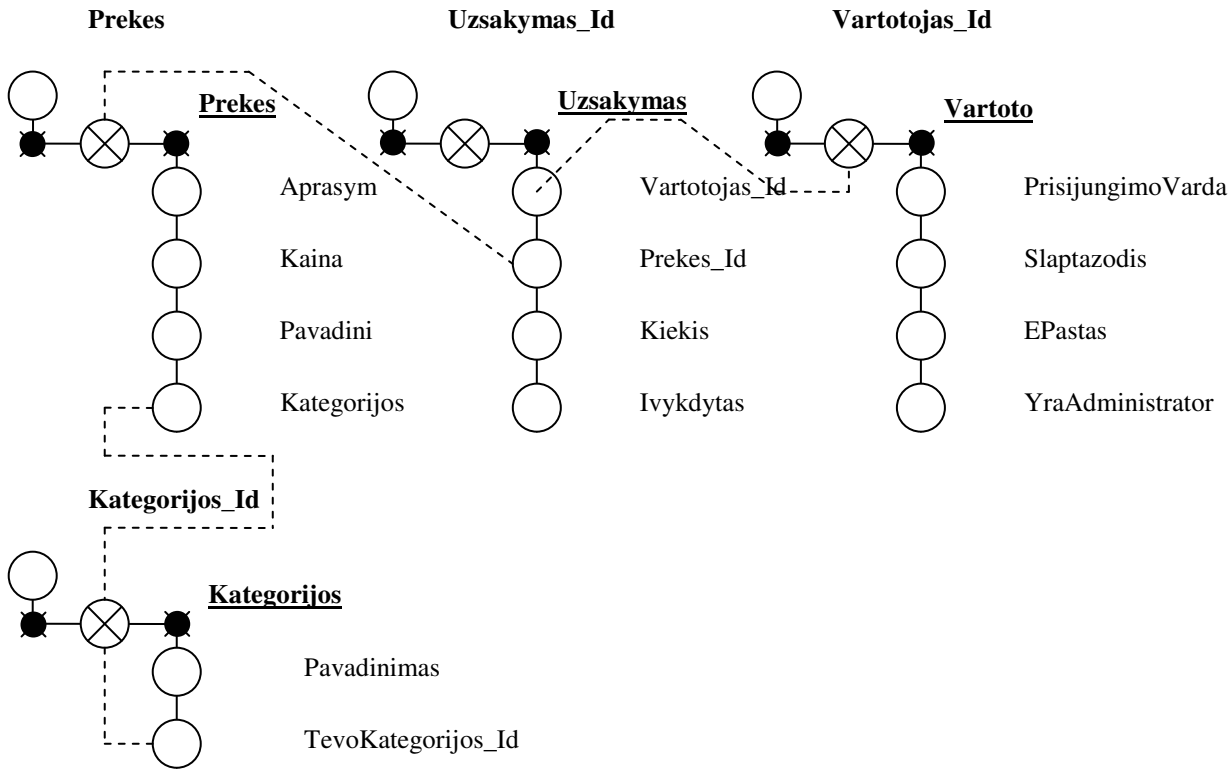
Prekes_Id	Pavadinimas Aprasymas Kaina

Kategorijos

Kategorijos_Id	Pavadinimas

3.20 pav. Kanoninė schema

3.11.5 Objektų-savybių modelis



3.21 pav. Objektų-savybių modelis

3.11.6 Duomenų bazės struktūra

```
# SQL DDL
# phpMyAdmin MySQL-Dump
# version 2.5.1
# http://www.phpmyadmin.net/ (download page)
#
# Host: localhost
# Generation Time: Dec 21, 2004 at 10:02 PM
# Server version: 3.23.56
# PHP Version: 4.3.3
# Database : `e_shop`
# --------------------------------------------------------

#
# Table structure for table `kategorijos`
#
# Creation: Dec 21, 2004 at 09:56 PM
# Last update: Dec 21, 2004 at 09:56 PM
#

DROP TABLE IF EXISTS kategorijos;
CREATE TABLE kategorijos (
 Kategorijos_Id int(11) NOT NULL default '0',
 Pavadinimas char(30) NOT NULL default '',
 TevoKategorijos_Id int(11) default NULL,
 PRIMARY KEY (Kategorijos_Id)
```

```
) TYPE=MyISAM;
# --------------------------------------------------------

#
# Table structure for table `prekes`
#
# Creation: Dec 21, 2004 at 09:58 PM
# Last update: Dec 21, 2004 at 09:58 PM
#

DROP TABLE IF EXISTS prekes;
CREATE TABLE prekes (
 Prekes_Id char(10) NOT NULL default '',
 Kategorijos_Id char(10) NOT NULL default '',
 Pavadinimas char(30) NOT NULL default '',
 Aprasymas char(100) default NULL,
 Kaina int(11) NOT NULL default '0',
 PRIMARY KEY (Prekes_Id)
) TYPE=MyISAM;
# --------------------------------------------------------

#
# Table structure for table `uzsakymas`
#
# Creation: Dec 21, 2004 at 09:59 PM
# Last update: Dec 21, 2004 at 09:59 PM
#

DROP TABLE IF EXISTS uzsakymas;
CREATE TABLE uzsakymas (
 Uzdakymas_Id int(11) NOT NULL auto_increment,
 Vartotojas_Id char(10) NOT NULL default '',
 Prekes_Id int(11) NOT NULL default '0',
 Kiekis int(11) NOT NULL default '0',
 Ivykdytas tinyint(4) NOT NULL default '0',
 PRIMARY KEY (Uzdakymas_Id)
) TYPE=MyISAM;
# --------------------------------------------------------

#
# Table structure for table `vartotojas`
#
# Creation: Dec 21, 2004 at 09:50 PM
# Last update: Dec 21, 2004 at 09:50 PM
#

DROP TABLE IF EXISTS vartotojas;
CREATE TABLE vartotojas (
 Vartotojas_Id int(11) NOT NULL auto_increment,
 PrisijungimoVardas char(10) NOT NULL default '',
 Slaptazodis char(10) NOT NULL default '',
 EPastas char(35) NOT NULL default '',
 YraAdministratorius tinyint(4) NOT NULL default '0',
 PRIMARY KEY (Vartotojas_Id)
) TYPE=MyISAM AUTO_INCREMENT=1 ;

# FOREIGN KEYS ADDED MANUALLY

ALTER TABLE prekes
      ADD CONSTRAINT Kategorijos_FK
      FOREIGN KEY(Kategorijos_Id)
      REFERENCES kategorijos(Kategorijos_Id);
```

```
ALTER TABLE uzsakymas
      ADD CONSTRAINT Vartotojas_FK
      FOREIGN KEY(Vartotojas_Id)
      REFERENCES vartotojas(Vartotojas_Id);

ALTER TABLE uzsakymas
      ADD CONSTRAINT Prekes_FK
      FOREIGN KEY(Prekes_Id)
      REFERENCES prekes(Prekes_Id)
```

3.12 Prototipo kūrimas – WAP terpė

Norint pademonstruoti esamo elektroninės komercijos sprendimo pritaikymą WAP terpei, prototipas iš pradžių buvo sukurtas vien WWW / HTML terpei, o vėliau modifikuojamas į daugiaterpį, kurio viena iš terpių – WAP / WML. Šiame skyriuje apibendrintai apžvelgsime kokie pakeitimai buvo atliekami sistemos pritaikymo WAP proceso metu.

3.12.1 Naujos terpės įtraukimas į sprendimą

Pirmas žingsnis paprastą elektroninės komercijos sprendimą modifikuojant į daugiaterpį, sprendimas kaip bus atskiriama kuriai terpei priklauso vartotojas, ir jei reikia – atskyrimo mechanizmo projektavimas ir programavimas.

Modifikuojant prototipą buvo sukurtas ir įdiegtas patogus, tačiau galimybių neapribojantis automatinis vartotojo agento nustatymas. Stacionarieji ir mobilieji įrenginiai (t.y. HTML naršyklės ir WML mikronaršyklės) jungiasi tuo pačiu įprastiniu sistemos adresu. Prisijungimo metu vartotojas nukreipiamas į atitinkamą turinį pagal jo naršyklės galimybes. Kas labiau tinka vartotojui – WML ar HTML, nusprendžiama iš jo naršyklės siunčiamų HTTP_ACCEPT pranešimų.

Naudojamas sekantis metodas automatiškai nustatyti pageidaujamo turinio tipą:

- **WML** - jeigu vartotojo naršyklės HTTP_ACCEPT (palaikomų duomenų tipų, priimamų HTTP protokolu sąrašas) aibėje yra tipas „text/vnd.wap.wml“ ir nėra tipo „text/html“
- **HTML** – kitais likusiais atvejais

Tačiau papildomai buvo sukurti adresai kuriais galima tiesiogiai pasiekti norimą sistemos terpę (HTML arba WML), „apeinant“ automatinį turinio tipo nustatymą. Šie adresai yra tie patys, į kuriuos vartotojas yra nukreipiamas automatinio atpažinimo metu. Tai yra naudinga tais atvejais, kai naršyklė palaiko ir WML ir HTML, o vartotojas nori matyti WML versiją (pavyzdžiui testavimo tikslais).

Norint kad pagal nutylėjimą elektroninės komercijos sprendimo adresu veiktų automatinis pageidaujamo turinio tipo nustatymas, reikia jog žiniatinklio serveris prisijungus naujam vartotojui, visų pirma pateiktų automatinio atpažinimo puslapį. Yra daug būdų kaip tai padaryti, ir jie priklauso nuo konkretaus žiniatinklio serverio. Lengviausias ir universaliausias būdas - automatinio atpažinimo puslapį pavadinti „index.html“ ir jį patalpinti į žiniatinklio serverio šakninį katalogą.

Kadangi elektroninės komercijos sprendime kurį modifikuojame šablonai nėra naudojami, yra du sprendimo būdai – arba juos pradėti naudoti, arba WAP prieigai reikia tiesiog sukurti naują programos arba atvaizdavimo sluoksnio dalį. Kadangi prototipas naudoja sluoksninę architektūrą, ir turi atvaizdavimo sluoksnį – sukursime dar vieną atvaizdavimo sluoksnį skirtą WAP terpei. Kad tai padaryti naudosime esamą WWW atvaizdavimo sluoksnį kaip pagrindą, ir jį versime naudodami anksčiau sukurtą HTML vertimo į WML metodiką.

3.12.2 Žymų vertimas

Kadangi prototipo kūrime buvo naudojamos scenarijų kalbos – neturime vientisų HTML dokumentų kuriuos galėtumėme versti WML dokumentus, o turime tik HTML fragmentus esančius PHP scenarijų kalboje. Prototipo kūrime taip pat nebuvo naudojami šablonai. Dėl šių priežasčių reikės naudoti rankinį HTML fragmentų esančių PHP scenarijuose (atvaizdavimo sluoksnyje) vertimą, remiantis anksčiau sukurta bei aprašyta HTML vertimo į WML metodika (nepamirštant jog taisyklės yra išplečiamos). Žemiau pateikiamas šio proceso pavyzdys bei rezultatas.

3.7 lentelė Supaprastintas atvaizdavimo sluoksnio metodo vertimas iš HTML į WML

HTML	WML
`// prints given items in HTML` `function printAll()` `{` `print "<table class=t1>";` `print "<tr>";` `print "<th>Pavadinimas</th>";` `print "<th>Kaina</th>";` `print "<th>Nuoroda</th>";` `print "</tr>";` `# output data rows` `foreach ($this->ITEMS as $line)` `{` `$title = $line['title'];` `$price = $line['price'];` `$id = $line['id'] ;` `print "<tr>` `print "<td>$title</td>` `print "<td>$price<td>` `print "<td>“` `print "<a href='item.php?id=$id'>>></a>` `print "</td>` `print "</tr>";` `}` `print "</table>";` `}`	`// prints given items in WML` `function printAll()` `{` `# output data rows` `foreach ($this->ITEMS as $line)` `{` `$title = $line['title'];` `$price = $line['price'];` `$id = $line['id'] ;` `print "$title - $price ";` `print "<a href='item.php?id=$id'>";` `print ">></a> ";` `}` `}`

3.8 lentelė Atvaizdavimo sluoksnio metodų sugeneruotų rezultatų pavyzdys

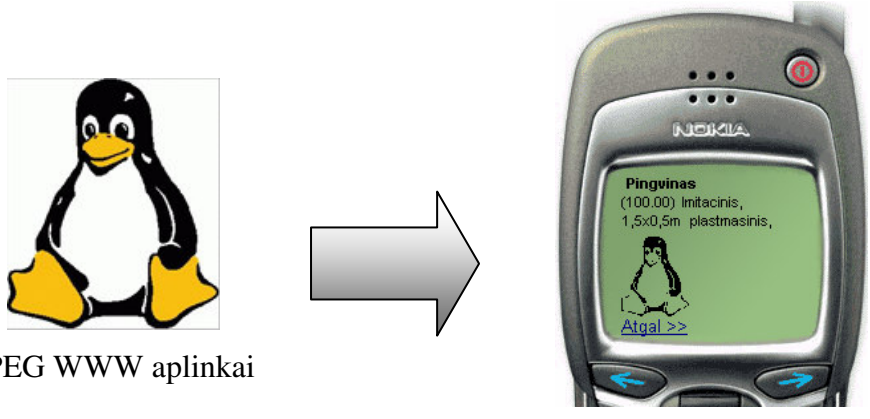
HTML	WML
<table class=t1> <tr> <th>Pavadinimas</th> <th>Kaina</th> <th>Nuoroda</th> </tr> <tr> <td>Preke1</td> <td>11<td> <td>>></td> </tr> <tr> <td>Preke2</td> <td>22<td> <td>>></td> </tr> <tr> <td>Preke3</td> <td>33<td> <td>>></td> </tr> </table>	Preke1 - 11 >> Preke2 - 22 >> Preke3 - 33 >>
Viso 344 baitai.	Viso 145 baitai.

Iš paskutinės lentelės matome jog iš HTML į WML išverstas ir supaprastintas dokumento fragmentas sumažėjo ~2,4 karto.

3.12.3 Grafinių elementų vertimas

Elektroninės parduotuvės prototipo statinių grafikos elementų kūrimas buvo atliktas rankiniu būdu, naudojant specialią programinę įrangą. Toks sprendimas buvo priimtas todėl, kad šių grafikos elementų paprastai yra nedaug, ir jie ilgai nesikeičia, todėl prasminga juos sukurti kuo kokybiškesnius.

Tačiau būtų labai nepatogu, jei tai reikėtų daryti kaskart įdėjus naują prekę į elektroninę parduotuvę, ar norint pakeisti jos esamą atvaizdą. Todėl buvo suprogramuota sistemos dalis, kuri kiekvieną naują įdėtą prekės atvaizdą automatiškai paversdavo ir į bevieliams įrenginiams tinkantį 60x60 taškų dydžio WBMP nespalvotą atvaizdą. Automatinio grafinių elementų vertimo, rezultato pavyzdys pateiktas 3.22 pav.



JPEG WWW aplinkai

WBMP WAP aplinkoje

3.22 pav. Automatinio grafinių elementų vertimo, rezultato pavyzdys

4 ELEKTRONINĖS KOMERCIJOS SPRENDIMO PROTOTIPO PRITAIKYMO WAP TERPEI REZULTATŲ TYRIMAS

Šiame skyriuje bus atliktas rezultatų, kuriuos gavome atlikę WWW žiniatinklio elektroninės komercijos sprendimo prototipo pritaikymą WAP terpei, tyrimas trim svarbiausiais aspektais – bus atsakyta į klausimą ar tai įmanoma, kokia funkcionalumo dalis buvo prarasta, bei kokie padariniai vartotojo sąsajai.

4.1 Principinė galimybė pritaikyti elektroninės komercijos sprendimą WAP

Šiame darbe buvo sėkmingai sukurtas daugialypę prieigą turinčio elektroninės komercijos sprendimo prototipas (elektroninė parduotuvė). Tai buvo pasiekta modifikuojant egzistuojantį, įprastai WWW žiniatinklio sričiai skirtą elektroninės komercijos sprendimą-prototipą. Sprendimas turi prieigą tiek įprastiems ir delniniams kompiuteriams, tiek ir labai ribotas atvaizdavimo bei apribojimo galimybes turintiems mobiliesiems telefonams.

Todėl galime daryti išvadą jog pavyko įrodyti principinę galimybę transformuoti vidutinio sudėtingumo elektroninės komercijos sprendimą iš WWW žiniatinklio, į ribotų galimybių belaidžius įrenginius palaikantį WAP / WML. Tiesa tam tikros sudėtingesnės sistemos funkcijos į WAP perkeltos nebuvo (plačiau apie tai sekančiame skyriuje).

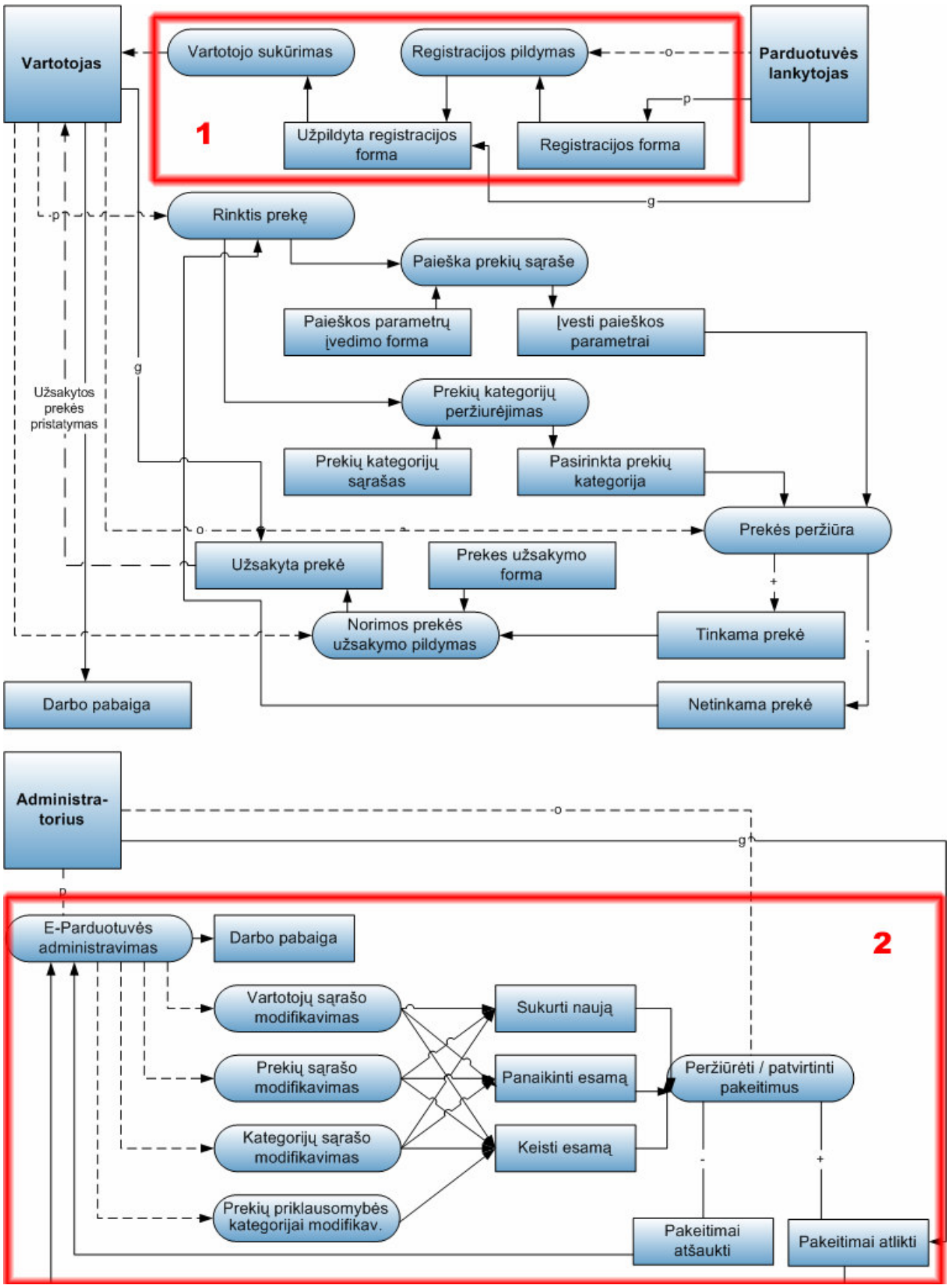
4.2 Funkciniai pokyčiai elektroninės komercijos sprendimą pritaikius WAP

Masiškai parduodami maži belaidžiai įrenginiai (pavyzdžiui, mobilieji telefonai, delniniai kompiuteriai) lyginant su įprastais kompiuteriais turi labai ribotus resursus. Dėl šių esminių energijos, dydžio ir įperkamumo apribojimų mobilieji įrenginiai dažniausiai turi:

- Mažesnio galingumo procesorius
- Mažiau atminties (RAM ir ROM)
- Ribojamą energijos suvartojimą
- Mažesnius ekranus
- Skirtingus ir ne tokius patogius įvesties įrenginius (pvz. mobiliojo telefono mygtukus)

Dėl šių priežasčių WAP aplinka ir jos galimybės gerokai skiriasi nuo WWW. Reikia įvertinti ir tai, jog netikslinga į WAP dėti tokį turinį, kurio nepalaikys dalis mobiliųjų įrenginių (net jei atsižvelgiant į WAP specifikacijas turėtų palaikyti). Taip pat netikslinga į WAP talpinti tokį turinį, kuris vartotojui bus pernelyg sudėtingas, ar nepatogus – tokiu atveju yra didelė tikimybė, jog šis turinys visai nebus naudojamas.

Sekančiame paveiksle raudonai pažymėtas funkcionalumas, kuris nebuvo įtrauktas į elektroninės komercijos sprendimą, jį pritaikius iš WWW į WAP.



4.1 pav. Organizacijos modelis WAP atvejui

Funkcionalumas, paveiksle pažymėtas „1“ – naujo vartotojo registracija. Įtraukti šį funkcionalumą į WAP sprendimą buvo netikslinga, nes registracijos patvirtinimui reikalingas

elektroninis paštas. Registracija skaitoma sėkmingai užbaigta tik tada, kai vartotojas nurodęs teisingą savo elektroninio pašto adresą, ir gavęs registracijos laišką, paspaudžia ant jame esančios nuorodos. Taip pat registruojantis reikia įvesti nemažai duomenų. Kadangi WAP atveju visa tai būtų sudėtinga ir nepatogu – dauguma vartotojų to tiesiog nedarytų. Todėl šis funkcionalumas veikia tik WWW elektroninės parduotuvės prieigoje. Tačiau vartotojai prisiregistravę WWW, gali sėkmingai naudotis visu likusiu parduotuvės funkcionalumu ir WAP prieigoje, o neprisiregistravę tiesiog neturės teisės atlikti prekės užsakymą.

Funkcionalumas, paveiksle pažymėtas „2“ – parduotuvės administravimas. Įtraukti šį funkcionalumą į WAP sprendimą taip pat buvo netikslinga, nes viso funkcionalumo neįmanoma įdiegti dėl WAP ir atskirų įrenginių apribojimų (pvz. nepavyktų įdėti naujų nuotraukų, nes iš mobiliojo telefono jų išsiųsti nepavyktų). Taip pat buvo atsižvelgta ir į tai, jog parduotuvės administravimas per mobilųjį įrenginį nebūtų patogus, ir taip pat būtų itin retai naudojamas.

Išsamesnė informacija apie pagrindinio funkcionalumo pokyčius WAP taikyme pateikta sekančioje lentelėje 4.1

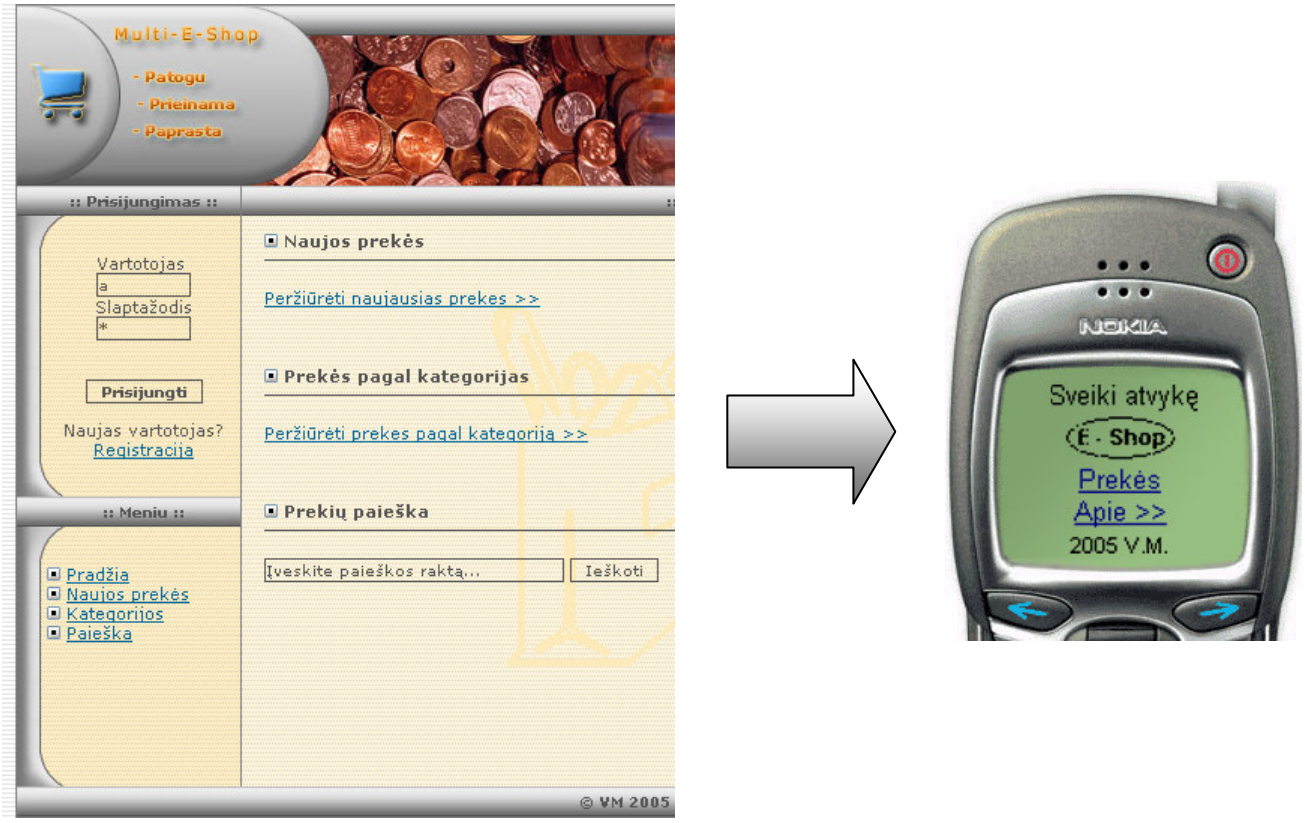
4.1 lentelė Funkcionalumo pokyčiai WAP taikyme

Funkcionalumas	Funkcionalumo kategorija	Pokyčiai WAP aplinkai
Registracijos pildymas	Vartotojo registracija	WAP taikyme šio funkcionalumo nėra dėl jo netikslingumo belaidei aplinkai.
Vartotojo sukūrimas	Vartotojo registracija	WAP taikyme šio funkcionalumo nėra dėl jo netikslingumo belaidei aplinkai.
Paieška prekių sąraše	Prekės pasirinkimas	Ši funkcija WAP taikyme veikia analogiškai WWW atitikmeniui. Tačiau skiriasi gražinamų rezultatų detalumas – norint sumažinti siunčiamų duomenų kiekį (dėl našumo ir navigacijos patogumo) dalis informacijos nerodoma, kol vartotojas nepasirenka tam tikros prekės iš pateikto sąrašo.
Prekių kategorijų peržiūra	Prekės pasirinkimas	Ši funkcija WAP taikyme taip pat veikia analogiškai WWW atitikmeniui. Tačiau ir čia skiriasi kataloge rodomų prekių informacijos detalumas – dalis informacijos nerodoma, kol vartotojas nepasirenka tam tikros prekės iš pateikto sąrašo.
Prekės peržiūra	Prekės pasirinkimas	Šis funkcionalumas yra tik WAP taikyme. Jis leidžia sumažinti perduodamų duomenų kiekį rodant prekių sąrašą įvairiose elektroninės parduotuvės dalyse (nes tada nebūtina rodyti pilną kiekvienos prekės informaciją – užtenka nuorodos į šį, prekės peržiūrėjimo, funkcionalumą). WWW atveju pilna informacija pateikiama jau prekių sąraše, o nuoroda naudojama tik prekių nuotraukoms.
Prekės užsakymas	Prekės užsakymas	Šis funkcionalumas buvo piltai įgyvendintas WAP aplinkoje. Pagrindinis skirtumas tarp WAP ir WWW sprendimo yra tas, kad WAP atveju vartotojo prisijungimo duomenis reikia įvesti tik prieš užsakant prekę, o WWW atveju tai galima padaryti bet kada.
Vartotojų sąrašo	Parduotuvės	WAP taikyme šio funkcionalumo nėra dėl jo netikslingumo

Funkcionalumas	Funkcionalumo kategorija	Pokyčiai WAP aplinkai
modifikavimas	administravimas	belaidei aplinkai.
Prekių sąrašo modifikavimas	Parduotuvės administravimas	WAP taikyme šio funkcionalumo nėra dėl jo netikslingumo belaidei aplinkai.
Kategorijų sąrašo modifikavimas	Parduotuvės administravimas	WAP taikyme šio funkcionalumo nėra dėl jo netikslingumo belaidei aplinkai.
Prekių priklausomybės kategorijai modifikavimas	Parduotuvės administravimas	WAP taikyme šio funkcionalumo nėra dėl jo netikslingumo belaidei aplinkai.

4.3 Vartotojo sąsajos pokyčiai elektroninės komercijos sprendimą pritaikius WAP

Atlikus WWW žiniatinklio elektroninės komercijos sprendimo prototipo pritaikymą WAP, galimas ne tik funkcionalumo prarandamas. Neabejotinai nukenčia ir vartotojo patyrimas / pasitenkinimas naudojantis sistema. Vietoje „gyvos“, gerai išdėstytos ir patogios vartotojui WWW žiniatinklio sprendimo sąsajos, gaunama gerokai nepatogesnė ir mažiau patraukli WAP / WML įrenginio sąsaja. Skirtumus akivaizdžiai galima pamatyti pradinio elektroninės parduotuvės puslapio skirto WAP ir WML terpėms atvaizduose, kurie pateikti paveiksle 4.2.



4.2 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – pradinis puslapis

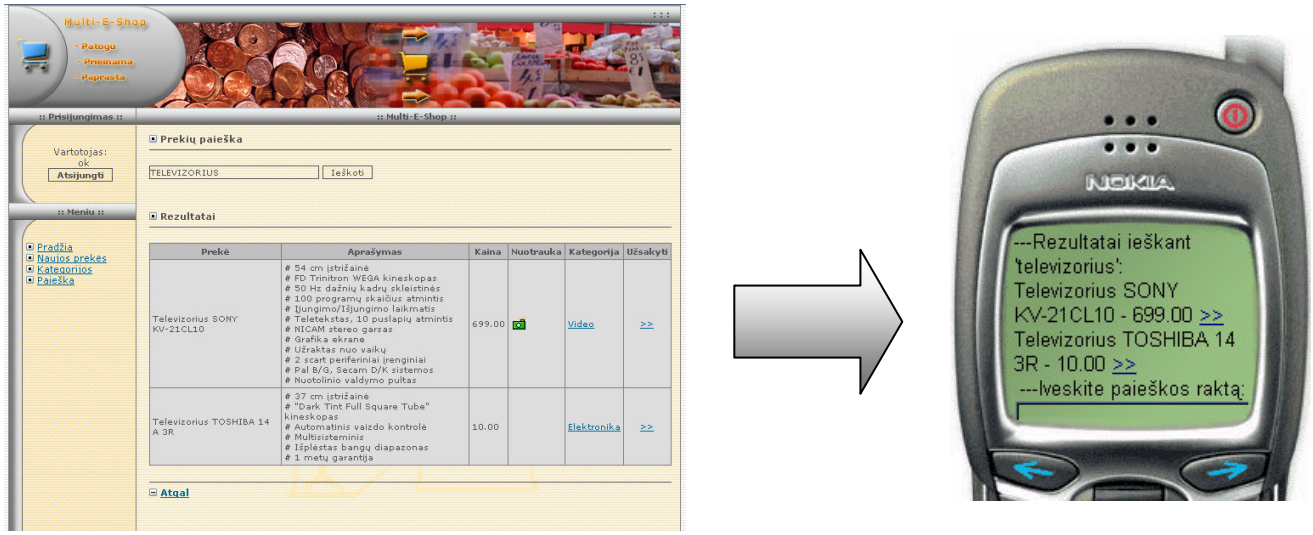
Išsamesnė informacija apie pagrindinius vartotojo sąsajos pokyčius WAP taikyme pateikta sekančioje lentelėje 4.2

4.2 lentelė Vartotojo sąsajos pokyčiai WAP taikyme

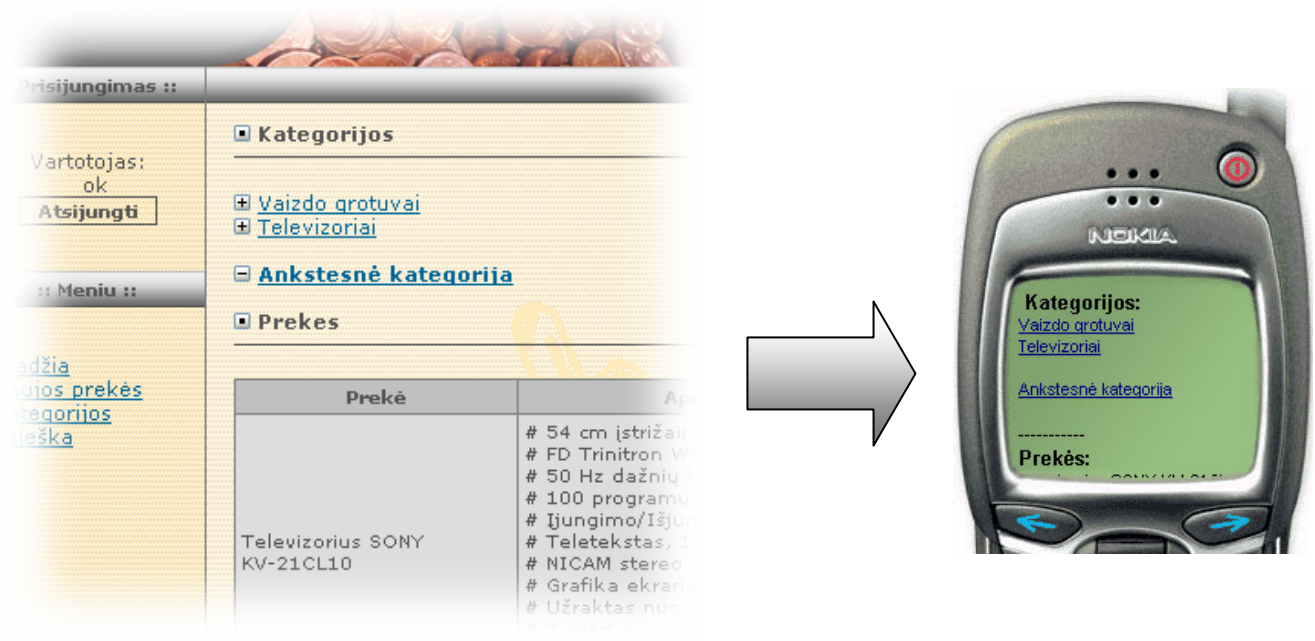
Sąsajos vieta	Pokyčiai WAP aplinkai
Paieška prekių sąraše	Rezultatuose nerodoma dalis informacijos, tačiau i ją pateikiama nuoroda. Nėra meniu nuorodų į kitas el. parduotuvės dalis.
Prekių kategorijų peržiūra	WAP taikyme vartotojo sąsaja beveik analogiška WWW atitikmeniui. Vėlgi nerodoma dalis informacijos, tačiau i ją pateikiama nuoroda. Nėra meniu nuorodų į kitas el. parduotuvės dalis.
Prekės peržiūra	Šis funkcionalumas yra tik WAP taikyme, nes WWW atveju pilną informaciją apie prekę galime matyti jau prekių sąraše (o nuoroda naudojama tik nuotraukoms).
Prekės užsakymas	Vartotojo sąsajos pokyčiai minimalūs. WAP taikyme nėra meniu nuorodų į kitas el. parduotuvės dalis, tik į tą, iš kurios buvo užsakyta prekė. Taip pat WAP taikyme prisijungimas vyksta prekės užsakymo metu.

Visi 4.2 lentelėje pateikti pakeitimai (pritaikymai) buvo atlikti dėl ribotų WAP įrenginių, ir jų turimų mikronaršyklių galimybių, o taip pat ir stengiantis naudojimąsi sistema padaryti kuo patogesniu. Kiekviename puslapyje pateikiamos informacijos kiekis buvo mažinamas tam kad puslapiai užsikrautų greičiau, o juose rasti reikalingą informaciją būtų lengviau.

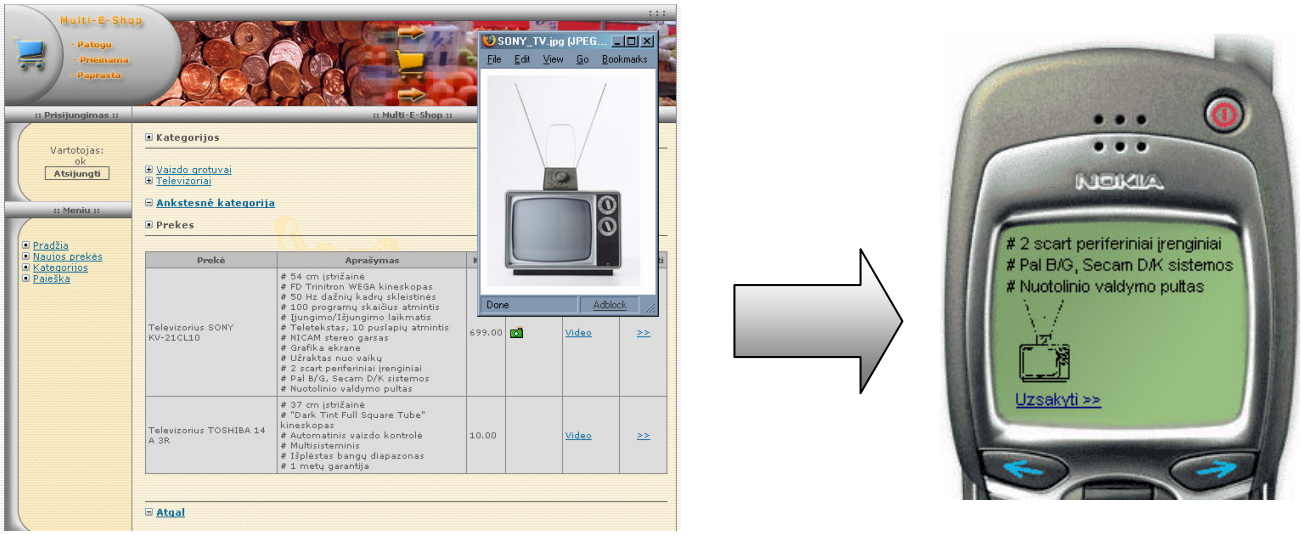
Sekančiuose paveiksluose (pav. 4.3 – 4.6) pateikiami paskutinėje lentelėje (lentelė 4.2) aprašytų vartotojo sąsajos pakeitimų atvaizdai.



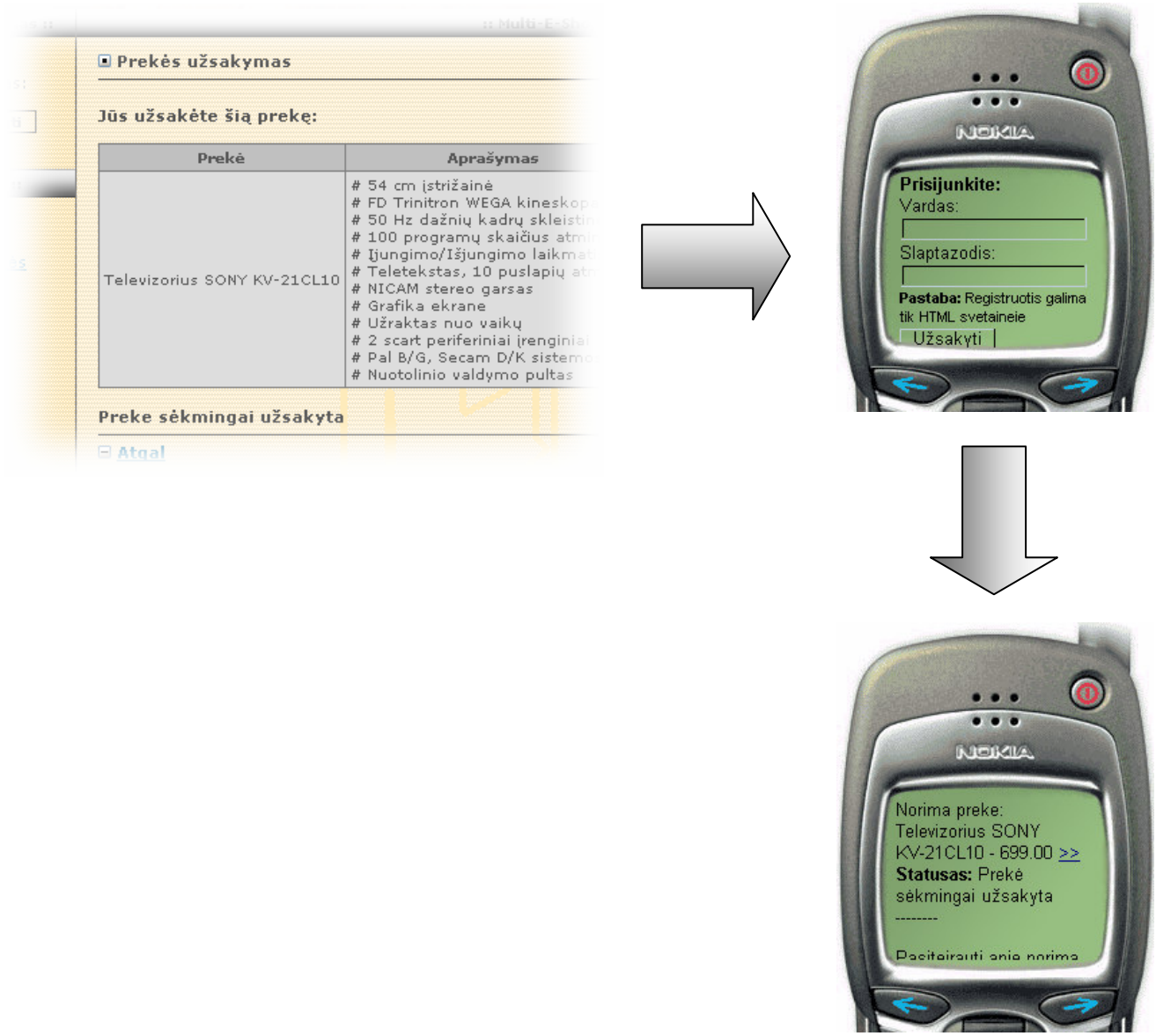
4.3 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – paieška prekių sąraše



4.4 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – prekių kategorijų peržiūra



4.5 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – prekės peržiūra



4.6 pav. Vartotojo sąsajos pasikeitimas iš WWW į WAP – prekės užsakymas

Tačiau be mažų mobiliųjų įrenginių sukeliamų nepatogumų (ribotos atvaizdavimo, apdorojimo bei įvesties galimybės) yra akivaizdūs ir dideli jų privalumai. Tokį WAP elektroninės komercijos sprendimą galima pasiekti iš bet kur ir bet kada - nereikia ieškoti įprasto kompiuterio, arba sustoti vien tam, kad išsitraukti bei įsijungti nešiojamąjį kompiuterį.

5 IŠVADOS

Šiame darbe buvo pristatyta galima WWW žiniatinklio elektroninės komercijos sprendimų pritaikymo WAP sričiai metodika ir taisyklės. Naudojantis šia metodika bei taisyklėmis, buvo sukurtas daugialypę prieigą turinčio elektroninės komercijos sprendimo prototipas (elektroninė parduotuvė). Sprendimas turi prieigą tiek įprastiems ir delniniams kompiuteriams, tiek ir labai ribotas atvaizdavimo bei apdorojimo galimybes turintiems mobiliesiems telefonams.

Galima teigti, jog svarbiausi darbo rezultatai yra šie:

1. Įrodyta principinė galimybė transformuoti vidutinio sudėtingumo elektroninės komercijos sprendimą, iš WWW žiniatinklio į ribotų galimybių belaidžius įrenginius, panaudojant WAP / WML technologijas (šios transformacijos metu galimas dalies sudėtingesnio funkcionalumo prarandamas).

2. Buvo pristatyta WWW žiniatinklio elektroninės komercijos sprendimų pritaikymo WAP sričiai metodologija, įskaitant:

    a) galimų sprendimo „pernešimo“ iš WWW į WAP architektūrų sąrašas, pateikiant kiekvienos architektūros privalumus bei trūkumus.

    b) išplečiamos taisyklės HTML vertimui į WML.

3. Sukurtas daugialypę prieigą turintis elektroninės komercijos sprendimo prototipas.

6 LITERATŪRA

[1] ACHOUR, M.; BETZ, F.; DOVGAL, A. *PHP manual* [interaktyvus] PHP: Hypertext preprocessor. 2005, [žiūrėta 2004 kovo 12]. Prieiga per internetą: <http://www.php.net/manual/en/>.

[2] *MySQL Reference Manual* [interaktyvus] MySQL. [žiūrėta 2004 vasario 02]. Prieiga per internetą: <http://www.mysql.com>.

[3] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. The Unified Modeling Language User Guide. Addison Wesley, 2000.

[4] RAGGETT, D. *HTML 4.01 Specification* [interaktyvus] World Wide Web Consortium. Recommendation 24 December 1999. 1999 gruodis, [žiūrėta 2005 sausio 6]. Prieiga per internetą: <http://www.w3.org/TR/html401/>.

[5] BRAY, T. *Extensible Markup Language (XML) 1.1* [interaktyvus] World Wide Web Consortium. Recommendation 04 February 2004. 2004 vasaris, [žiūrėta 2005 vasario 6]. Prieiga per internetą: < http://www.w3.org/TR/2004/REC-xml11-20040204/>.

[6] BYRNE, S. *Document Object Model (DOM) Level 3 Core Specification* [interaktyvus] World Wide Web Consortium. Recommendation 07 April 2004. 2004 balandis, [žiūrėta 2005 balandžio 6]. Prieiga per internetą: <http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407/>.

[7] *Wireless Application Protocol Architecture* [interaktyvus] Wireless Application Protocol Forum. [žiūrėta 2004 sausio 02]. Prieiga per internetą: <http://www.wapforum.org/what/technical_1_2_1.htm>.

[8] *Wireless Application Environment Overview* [interaktyvus] Wireless Application Protocol Forum. [žiūrėta 2004 sausio 02]. Prieiga per internetą: <http://www.wapforum.org/what/technical_1_2_1.htm>.

[9] *General Formats Specification* [interaktyvus] Wireless Application Protocol Forum. [žiūrėta 2004 sausio 02]. Prieiga per internetą: <http://www.wapforum.org/what/technical_1_2_1.htm>.

[10] *TCP/IP specification* [interaktyvus] RFC. [žiūrėta 2004 lapkričio 02]. Prieiga per internetą: <http://rfc.net/rfc793.html>.

[11] FIELDING, R. *Hypertext Transfer Protocol* 1.1 [interaktyvus] RFC. [žiūrėta 2004 sausio 02]. Prieiga per internetą: <http://rfc.net/rfc2616.html>.

[12] FIELDING, R. *Uniform Resource Identifiers* [interaktyvus] RFC. [žiūrėta 2004 liepos 02]. Prieiga per internetą: <http://rfc.net/rfc2396.html>.

[13] FREED N.; BORENSTEIN N. *Multipurpose Internet Mail Extensions* [interaktyvus] RFC. [žiūrėta 2004 rugsėjo 07]. Prieiga per internetą: <http://rfc.net/rfc2045.html>.

[14] *HTTP Client Authentication*. [interaktyvus] RFC. [žiūrėta 2004 balandžio 09]. Prieiga per internetą: <http://rfc.net/rfc2617.html>.

[15] KENT, S. *Security Architecture for the Internet Protocol* [interaktyvus] RFC. [žiūrėta 2005 sausio 02]. Prieiga per internetą: <http://rfc.net/rfc2401.html>.

[16] *MOCKAPETRIS, P. Domain Names – Concepts and Facilities*. [interaktyvus] _. [žiūrėta 2005 vasario 12]. Prieiga per internetą: <http://rfc.net/std13.html>.

[17] POSTEL, J. *User Datagram Protocol* [interaktyvus] RFC. [žiūrėta 2004 gegužės 11]. Prieiga per internetą: <http://rfc.net/rfc768.html>.

[18] *IP internet protocol specification*. [interaktyvus] RFC. [žiūrėta 2005 sausio 07]. Prieiga per internetą: <http://rfc.net/rfc791.html>.

[19] DIERKS, T. The TLS protocol version 1.0 [interaktyvus] RFC. [žiūrėta 2004 kovo 06]. Prieiga per internetą: <http://rfc.net/rfc2246.html>.

[20] *Wireless profiled TCP Specification* [interaktyvus] OMA. [žiūrėta 2004 gegužės 04]. Prieiga per internetą: <http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html>.

[21] DEERING, S. *IP internet protocol specification, version 6 (IPv6)*. [interaktyvus] RFC. [žiūrėta 2005 sausio 07]. Prieiga per internetą: <http://rfc.net/rfc2460.html>.

7 TERMINŲ IR SANTRUMPŲ ŽODYNAS

Žemiau pateiktos dažniausiai darbe naudotos santrumpos bei terminai:

- EFI (*External Functionality Interface*) – išorinio funkcionalumo sąsaja.
- GPRS (*General Packet Radio Service*) – bendra paketų pardavimo radijo tinklais paslauga
- GSM (*Global System for Mobile communications*) – globali mobiliojo ryšio sistema.
- HTTP (*Hypertext Transport Protocol*) – protokolas, naudojamas dokumentų perdavimui pasauliniame žiniatinklyje.
- LEAP (*Lightweight and Efficient Application Protocol*) – lengvasvoris ir efektyvus taikymų protokolas.
- MySQL – Duomenų bazių valdymo sistema.
- MMS (*Multimedia Messaging Service*) – dauglialypės terpės pranešimų paslauga.
- PHP (*Hypertext Pre-processor*) – programavimo kalba.
- RUP (*Rational Unified Process*) – programinės įrangos kūrimo (projektavimo, testavimo ir kt.) procesas.
- SMS (*Short Message Service*) – trumpųjų pranešimų paslauga.
- TCP/IP (*Transmission Control Protocol/Internet Protocol*) – standartinis tinklo protokolas, leidžiantis kompiuteriams pasiekti internetą.
- UDP (*User Datagram Protocol*) – vartotojo datagramų protokolas.
- UML (*Unified Modelling Language*) – universali modeliavimo kalba.
- URI (*Unified Resource Identifier*) – visuotinis resurso identifikatorius.
- URL (*Uniform Resource Locator*) – visuotinis resurso radėjas
- W3C (*WWW Consortium*) – pasaulinio žiniatinklio konsorciumas.
- WAE (*WAP Application Environment*) – WAP taikymų aplinka.
- WAP (*Wireless Application Protocol*) – belaidis taikymų protokolas.
- WDP (*Wireless Datagram Protocol*) – belaidis datagramų protokolas.
- WIM (*Wireless Identity Module*) - belaidžio identifikavimo modulis.
- WML (*Wireless Markup Language*) – belaidė žymų kalba.
- WSP (*Wireless Session Protocol*) – belaidžių sesijų protokolas.
- WTLS (*Wireless Transport Layer Security*) – belaidžio transporto sluoksnio saugumas.
- WTP (*Wireless Transaction Protocol*) – belaidžių transakcijų protokolas.
- XML (*Extensible Markup Language*) – išplėstoji žymių kalba.

8 PRIEDAI

8.1 WAP protokolo galimybių analizė

8.1.1 Įžanga į WAP

WAP (*Wireless Application Protocol*) yra didelis žingsnis į priekį, suteikiantis universalią, Internetu pagrįstą informacijos prieigą belaidžiams įrenginiams. Jis įgalina programinės įrangos kūrėjams kurti vieną kartą – visiems pasaulio tinklams. Nešėjai (pvz. dabar paplitęs – GSM) gali sukurti vartus (angl. *gateway*) kurie naudojantis įvairiais telefonais leis naudotis visomis tinklo taikomosiomis programomis bei turiniu. Telefonų bei delninių kompiuterių gamintojams tai leis gaminti nebrangius įrenginius visiems nešėjams.

WAP (angl. *Wireless Application Protocol*) paskirtis yra suteikti operatoriams, techninės įrangos gamintojams, programinės įrangos bei turinio kūrėjams bendrą aplinką, kuri leis kurti bei vystyti papildomos vertes paslaugas mobiliesiems įrenginiams. Keturi WAP įkūrėjai, dar vadinami (Ericsson, Motorola, Nokia ir Unwired planet) stengiasi su kitais partneriais sukurti globalias belaidžių paslaugų specifikacijas, nusakančias griežtus standartus, tačiau nepriklausomas nuo vietinio nešančiojo tinklo, bei jo struktūros. Visos protokolo savybės yra panaudojamos nepriklausomai nuo nešančiojo tinklo, bei įrenginių tipų. Organizacija kuri rūpinasi WAP standartais, bei yra jų savininkė (dabar jau valdoma Open Mobile Alliance) yra „WAP Forum“.

WAP (angl. *Wireless Application Protocol*) siekiama teikti pažangias paslaugas ir interneto turinį mobiliesiems telefonams bei kitiems nedideliems įrenginiams. Bendras standartas lemia tai, kad didėja pardavimų skaičius, mobiliųjų įrenginių gamintojai investuoja į suderinamų įrenginių kūrimą, o belaidžių tinklų atstovus verčia teikti naujus, diferencijuotus paslaugų pasiūlymus, taip pritraukiant naujus vartotojus. Tuo tarpu vartotojai gauna naudą iš platesnio pažangaus mobiliojo komunikavimo panaudojimo bei platesnio paslaugų spektro.

8.1.1.1 WAP Forum siekiai

WAP (angl. *Wireless Application Protocol*) yra nenutrūkstamų pastangų sukurti specifikacijas, skirtas taikomųjų programų veikiančių belaidžiais telekomunikacijų tinklais, rezultatas. WAP Forum siekis yra sukurti specifikacijų aibę, kuri bus naudojama paslaugų kūrimui bei taikymui. „Belaidė“ rinka yra sparčiai auganti, pasiekianti naujus vartotojus bei siūlanti vis naujas paslaugas. Norint operatoriams bei įrangos gamintojams suteikti galimybę nugalėti sunkumus atsirandančius pažangių, įvairialypių, greitų ir lanksčių paslaugų teikime bei kūrime, WAP parenka bei aprašo aibę atvirų, išplečiamų protokolų bei turinio formatų, kaip šių įvairialypių (ypač įrangos prasme) paslaugų įgyvendinimo pagrindą.

WAP Forum tikslai yra:

- Padaryti prieinamais interneto turinį bei pažangias duomenų paslaugas mobiliesiems telefonams bei kitiem belaidžiams terminalams.
- Sukurti globalią belaidžio protokolo specifikaciją, kuri galės veikti įvairių technologijų belaidžiuose tinkluose.
- Įgalinti turinio ir taikomųjų programų, kurios gerai veiktų labai plačiame nešančiųjų tinklų bei „galutinio vartotojo“ įrenginių spektre, kūrimą.
- Priimti ir išplėsti standartus bei technologijas kai tik atsiras poreikis.

WAP architektūros specifikacija yra skirta pristatyti sistemą bei architektūras, būtinas norint pasiekti WAP Forum užsibrėžtus tikslus. WAP architektūros specifikacija yra pradinis taškas norint suprasti WAP technologijas bei visas kitas iš jos sekančias specifikacijas. Ši specifikacija apžvelgia skirtingas naudojamas technologijas, bei turi savyje nuorodas į tolesnes bei detalesnes specifikacijas.

Dabartinė WAP architektūros specifikacija tęsia pirminės specifikacijos kryptį bei pasisekimą. Tinklo elementų funkcionalumas lieka panašus. Pavyzdžiui, architektūra naudoja našumo didinimo, bei funkcionalumo gerinimo tarpinius serverius (*proxies*) tam, kad sumažinti apdorojimo reikalavimus apribotiems (greičiu, atminties kiekiu ir kt. parametrais) įrenginiams, atskleisti belaidžio tinklo galimybes ir funkcijas, bei suteikti tinklo bei paslaugų valdymo galimybes. Dabartinė WAP architektūros versija buvo pagerinta, ir suteikia platesnį prisijungimo kelių pasirinkimą tarp klientų bei serverių, kai to reikia, pavyzdžiui tam kad tiekti baigtis-baigtis (angl. *end-to-end*) saugumą.

Pati WAP architektūros specifikacija suteikia karkasą plačiam protokolų, galimybių ir paslaugų spektrui. Ji nenurodo jokių įgyvendinimo detalių ar specifikos, todėl yra daugiau informacinio pobūdžio.

8.1.2 WAP technologijos apžvalga

WAP yra pozicionuojamas į trijų greitai besivystančių tinklo technologijų susiliejimą: belaidžių duomenų perdavimo, telefonijos bei interneto.

Tiek belaidžių duomenų perdavimo rinka, tiek internetas vis dar labai greitai auga ir pasiekia vis naujus vartotojus. Staigus interneto „augimas“ paspartino naujų bei įdomių informacinių paslaugų kūrimą.

Dauguma šių naujai sukuriamų internetui skirtui technologijų yra skirtos staliniams ar galingesniems kompiuteriams, bei vidutinio arba didelio pralaidumo, patikimiems duomenų tinklams. Masiškai parduodami maži belaidžiai įrenginiai (tiek mobilieji telefonai tiek delniniai kompiuteriai) turi gerokai labiau apribotus resursus, lyginant su įprastais kompiuteriais. Dėl šių esminių energijos, dydžio bei įperkamumo apribojimų mobilieji įrenginiai yra linkę turėti:

- Mažesnio galingumo procesorius

- Mažiau atminties (RAM ir ROM)
- Ribojamą energijos suvartojimą
- Mažesnius ekranus
- Skirtingus įvesties įrenginius (pvz. mobiliojo telefono mygtukus)

Panaši situacija yra ir su belaidžiais tinklais – komunikavimo aplinka yra labiau apribota lyginant su paprastais tinklais. Dėl esminių energijos, turimo spektro, ir mobilumo, belaidžiai tinklai lyginant su paprastai yra line turėti:

- Mažesnį pralaidumą
- Didesnį vėlinimą
- Mažesnį sujungimo stabilumą
- Mažiau numatomą buvimą (*availability*)

Norint suteikti papildomos vertės paslaugas mobilieji tinklai darosi vis sudėtingesni, dėl ko kyla ir kaina norint teikti šias paslaugas. Norint atitikti mobiliųjų tinklų operatoriams taikomus reikalavimus, sprendimai turi būti:

- Suderinami – įvairių gamintojų terminalai (nesvarbu ar tai mobilus telefonas, ar delninis kompiuteris, ar kitas įrenginys) naudojasi mobiliojo tinklo paslaugomis
- Pritaikomi – mobiliųjų tinklų operatoriai gali pritaikyti paslaugas pagal klientų norus
- Efektyvūs – paslaugų kokybė pritaikyta mobiliojo tinklo veikimui ir charakteristikoms
- Patikimi – tieka pastovų ir numatomą pagrindą paslaugų tiekimui
- Saugūs – turi būti galimybė perduoti duomenis per neapsaugotus mobilius tinklus tačiau vartotojo duomenys turi išlikti nepakitę; taip pat įrenginiai ir paslaugos turi būti apsaugoti nuo saugumo problemų, tokių kaip konfidencialumo praradimas

Dauguma dabartinių mobiliųjų tinklų turi pažangias paslaugas kurios gali būti siūlomos vartotojams. Mobiliųjų tinklų operatoriai stengiasi tiekti pažangias paslaugas patogiu, lengvai naudojamu ir patraukliu būdu, tam kad šių paslaugų vartotojų būtų kuo daugiau, o jas atsisakančių – kuo mažiau. Standartinės galimybės, tokios kaip skambučių valdymas, gali būti pagerintos naudojant WAP technologiją – pavyzdžiui gali sukurtos ir pritaikytos vartotojų sąsajos. O tokios paslaugos kaip skambučio peradresavimas gali suteikti vartotojui galimybę rinktis ar atsiliepti į skambutį, ar peradresuoti jį kitam asmeniui, ar persiųsti į balso pašto dėžutę.

Belaidžių įrenginių pagrindinė ir būdingoji savybė yra ta, jog jie paprastai būna mobilieji. Mobilumas suteikia naujų galimybių paslaugoms kurios yra jautrios judėjimui, ir gali teikti informaciją priklausomą nuo buvimo vietos. WAP specifikacijos ir architektūra išskiria šį unikalų belaidžių įrenginių aspektą, įtraukdama mobilumą kaip savo taikymų modelio (angl. *application model*) dalį

WAP specifikacijos atkreipia dėmesį į mobiliųjų tinklų charakteristikas bei operatorių poreikius pritaikydama egzistuojančias tinklų technologijas specifiniams, masinio pardavimo mažų gabaritų belaidžių įrenginių, reikalavimams, o esant reikalui – pasiūlo ir naujas technologijas.

Taigi WAP specifikacijos pritaikytos plačiam įrenginių spektrui: nuo įrenginių kurie atlieka tik labai paprastas funkcijas (pvz. mobilieji telefonai), iki įrenginių kurių galimybės nepaliaujamai plečiasi (pvz. delniniai kompiuteriai). Tai yra motyvacija naudoti architektūrą, kuri leistų esant reikalui funkcionalumą perkelti į skirtingas tinklo dalis – t.y. arba į įrenginius, arba į tinklo darbines stotis, jei tokia būtinybė yra.

8.1.3 WAP Forum siekiami architektūros tikslai

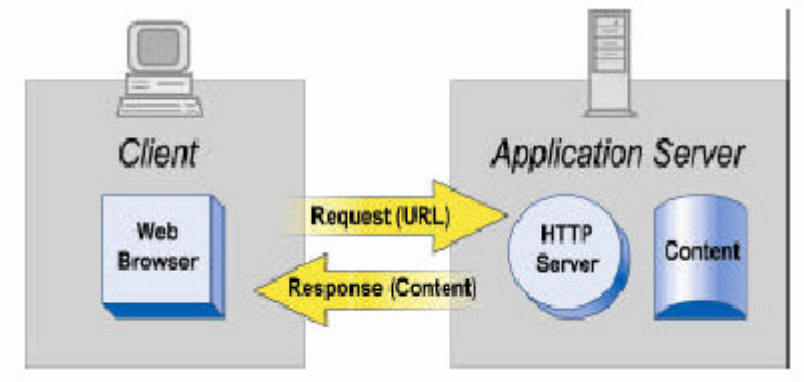
Šiame skyrelyje išvardinti WAP Forum siekiami architektūros tikslai. Ši santrauka yra informatyvi tačiau nėra labai detali. Išvardinti architektūros tikslai nėra surikiuoti, ir jų tvarka neatspindi prioriteto ar svarbumo.

- Tiekti žiniatinkliu (*web*) pagrįstą taikymų modelį (angl. *application model*) belaidėms duomenų paslaugoms, kurios naudoja telefoniją, mobilumą, bei kitas unikalias belaidžių įrenginių bei tinklų savybes. Tačiau tuo pačiu palieka maksimalų lankstumą, bei galimybę gamintojams suteikti vartotojams kuo geresnį vartojimo teikiamos naudos kiekį.
- Įgalinti įrenginio pritaikymą savo reikmėms (tinkinimą). Taip pat galimybė tinkinti gaunamą turinį, bei jo pateikimą.
- Palaikyti saugų ir neatskleidžiantį duomenų pašaliniams paslaugų teikimą bei komunikavimą, tuo pačiu išlaikant pastovumą ir atitikimą interneto saugumo modeliams.
- Palaikyti tiek dabartinius belaidžius įrenginius ir tinklus, tiek ir tuos kurie dar tik bus naudojami artimoje ateityje, įskaitant platų nešėjų spektrą – nuo siaurajuosčių iki plačiajuosčių.
- Tiekti saugų priėjimą prie vidinio įrenginio funkcionalumo.
- Palengvinti tinklo operatorių bei trečiųjų šalių teikiamų paslaugų tiekimą.
- Apibrėžti sluoksninę, pritaikomą pagal reikiamą mastą, bei išplečiamą architektūrą.
- Kai tai įmanoma – įtakoti esamus standartus, ypač esamus bei besivystančius interneto standartus.

8.1.4 WAP architektūros apžvalga

8.1.4.1 Pasaulinio žiniatinklio (*World Wide Web*) modelis

Pasaulinio interneto žiniatinklio (*WWW*) architektūra suteikia labai lankstų ir galingą programavimo modelį. Taikomosios programos bei turinys yra vaizduojamas standartiniais duomenų formatais, ir yra naršomas naudojant specialias programas, vadinamas žiniatinklio naršyklėmis. Žiniatinklio naršyklė yra tinklą naudojanti programa, t.y. ji siunčia užklausas tinklo stotims, įvardintiems duomenų objektams gauti, o tinklo serveris savo ruožtu atsako duomenimis, užkoduotais naudojant standartinius formatus.



Pasaulinio žiniatinklio programavimo modelis

Pasaulinio žiniatinklio (*WWW*) standartai nurodo daug mechanizmų, kurie yra būtini norint sukurti bendros paskirties programavimo aplinką, pvz.:

- Standartinis pavadinimų modelis (*Standard naming model*) – visi serveriai ir visas pasaulinio žiniatinklio turinys yra pavadinti naudojant interneto standartą URL (vieningas resurso radėjas – Uniform Resource Locator) [12]
- Turinio tipizavimas (*Content typing*) – Visas pasaulinio žiniatinklio turinys (tiksliau kiekvienas jo elementas) turi savo specifinį tipą, taip leisdamas žiniatinklio naršyklėms teisingai apdoroti turinį pagal jo tipą [13].
- Standartiniai turinio formatai (*Standard content formats*) – visos pasaulinio žiniatinklio naršyklės palaiko tam tikrą turinio formatų aibę. Tarp šių tipų yra:
  - Hiperteksto žymių kalba (*Hypertext Markup Language – HTML*) [4], scenarijų (*scripting*) kalbos *ECMAScript, JavaScript*, bei daug kitų formatų.
- Standartiniai protokolai (*Standard Protocols*) – standartiniai tinklinio darbo protokolai leidžia bet kuriai žiniatinklio naršyklei komunikuoti su bet kuriuo žiniatinklio serveriu. Plačiausiai naudojamas protokolas pasauliniame žiniatinklyje yra hiperteksto transportavimo protokolas (*Hypertext Transport Protocol - HTTP*) [11], veikiantis su TCP/IP protokolo rinkiniu. [10].

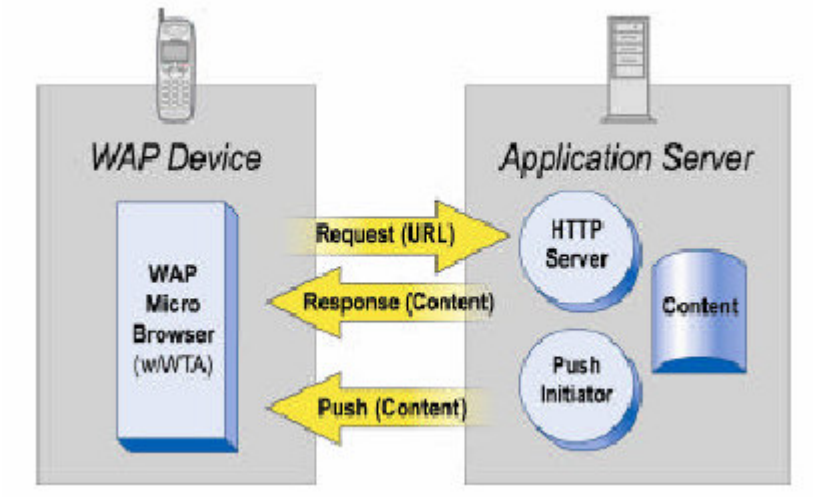
Ši infrastruktūra leidžia vartotojams lengvai pasiekti didelį trečios-šalies programų bei turinio paslaugų skaičių. Taip pat ji leidžia programinės įrangos kūrėjams lengvai kurti programas bei turinio paslaugas plačiai klientų bendruomenei.

8.1.4.2 WAP Modelis

WAP programavimo modelis yra tas pats pasaulinio žiniatinklio programavimo modelis su keletu patobulinimų. Pasaulinio žiniatinklio programavimo modelio pritaikymas suteikia papildomos naudos programinės įrangos kūrėjų bendruomenei, pvz. pažįstamas programavimo modelis, pasitvirtinusi architektūra, bei galimybė naudoti esamus įrankius (pvz. žiniatinklio serverius, XML įrankius ir kt.). Optimizacijos ir praplėtimai buvo daromi tam, kad pritaikyti WAP programavimo modelį belaidės aplinkos charakteristikoms. Visur kur įmanoma buvo pritaikyti egzistuojantys standartai arba jie buvo panaudoti kaip pradžios taškas WAP technologijai.

Pagrindiniai papildymai kurie buvo įdėti į WAP programavimo modelį yra:

- *Push*
- Telefonijos palaikymas (WTA)



WAP programavimo modelis

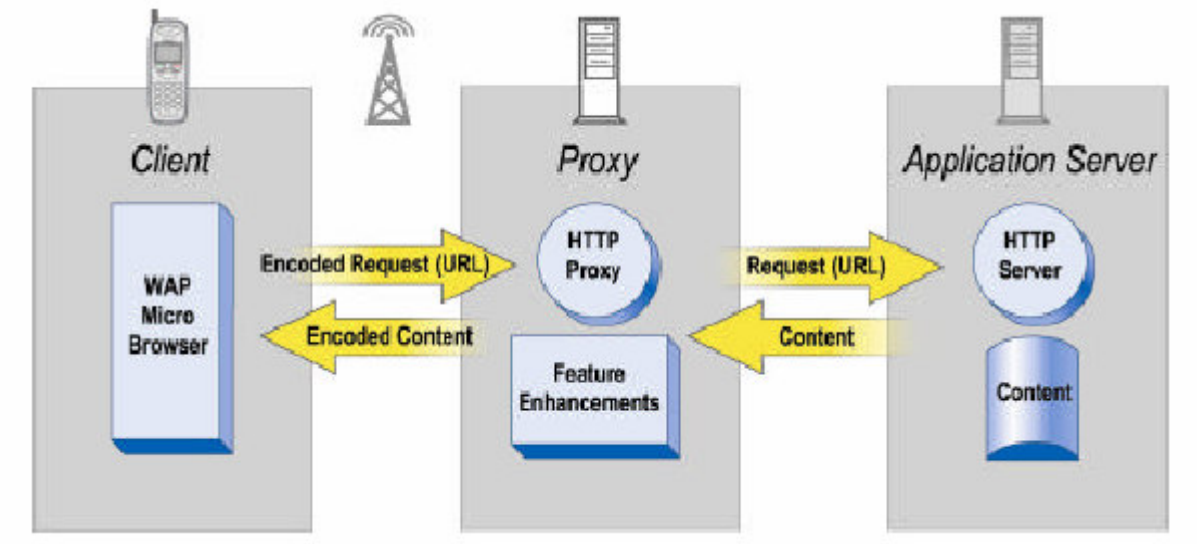
Klasikinis užklausos-atsakymo mechanizmas, dažnai dar vadinamas „*pull*“, skiriasi nuo „*push*“ mechanizmo. WAP turinys bei programos yra apibrėžtos aibe gerai žinomų turinio formatų, pagrįstų panašiais pasaulinio žiniatinklio (WWW) turinio formatais. Turinys yra transportuojamas naudojant standartinių komunikavimo protokolų aibę, kurie savo ruožtu taip pat yra pagrįsti panašiais pasaulinio žiniatinklio (WWW) komunikavimo protokolais. WAP mikro-naršyklė belaidžiame terminale koordinuoja vartotojo sąsają, ir yra analogas pasaulinio žiniatinklio naršyklei.

WAP apibrėžia standartinių komponentų aibę, kurie įgalina komunikavimą tarp mobiliųjų terminalų ir tinklo serverių, pvz.:

- Standartinis pavadinimų modelis (*Standard naming model*) – naudojamas pasaulinio žiniatinklio vieningas resurso radėjas (*Uniform Resource Locator - URL*) tam kad atpažinti WAP turinį serveriuose. Pasaulinio žiniatinklio vieningas resurso identifikatorius (*Uniform Resource Identificator - URI*) yra naudojamas identifikuoti vidinius įrenginio resursus, pvz. skambučių valdymo funkcijas.
- Turinio tipizavimas (*Content typing*) – Visas WAP turinys (tiksliau kiekvienas jo elementas) turi savo specifinį tipą, taip leisdamas WAP naršyklėms teisingai apdoroti turinį pagal jo tipą. WAP tipai paprastai atitinka egzistuojančius pasaulinio žiniatinklio tipus.
- Standartiniai turinio formatai (*Standard content formats*) – WAP turinio formatai yra pagrįsti pasaulinio žiniatinklio technologija, ir savyje turi atvaizdavimo žymių, kalendoriaus informacijos, elektroninių biznio kortelių objektus, paveikslus ir scenarijų kalbą.
- Standartiniai komunikavimo protokolai (*Standard communication protocols*) – WAP komunikavimo protokolai leidžia WAP naršyklei komunikuoti su žiniatinklio serveriu.

WAP turinio tipai ir protokolai yra optimizuoti masinei rinkai skirtiems nedidelių gabaritų bevieliams įrenginiams.

8.1.4.3 Savybes bei našumą pagerinantys tarpiniai serveriai (*proxies*)



Savybes bei našumą pagerinantys tarpiniai serveris

WAP naudoja tarpinio serverio technologiją tam kad pagerinti ir optimizuoti sujungimą tarp belaidės srities ir pasaulinio žiniatinklio.

Taigi WAP tarpiniai serveriai gali atlikti įvairias funkcijas, pavyzdžiui:

- Protokolo vartai (*Protocol Gateway*) – protokolo vartai paverčia užklausas iš belaidžio protokolo steko (pvz. WAP 1.x steko – WSP, WTP, WTLS ir WDP) į pasaulinio

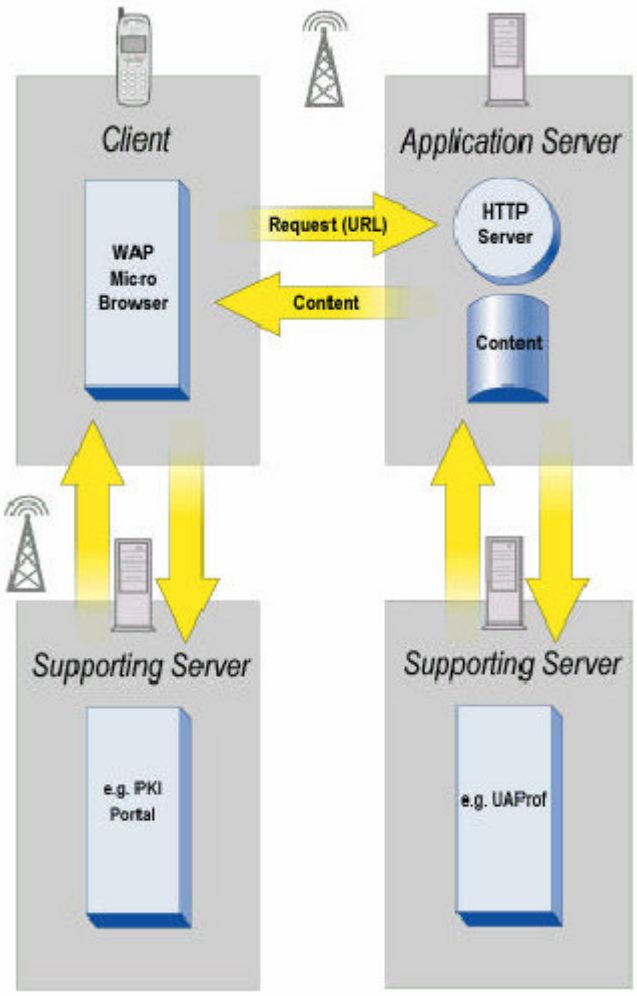
žiniatinklio protokolus (HTTP ir TCP/IP). Protokolo vartai taip pat atlieka DNS paieškas – ieškomi serveriai pagal kliento nurodytą URL (*Uniform Resource Locator*).

- Turinio užkodavimas ir dekodavimas – turinio užkodavimas gali būti naudojamas tam, kad išversti WAP turinį į kompaktišką formatą, kuris leidžia geriau išnaudoti (taupyti) ryšio kanalą dėl sumažėjusio duomenų dydžio.
- Vartotojo agento profilio valdymas (*User Agent Profile Management*) – vartotojo agento profiliai, nusakantys kliento galimybes ir asmeninius nustatymus, yra sukuriami ir pristatomi taikomosioms programoms.
- Spartinantysis tarpinis serveris (*caching proxy*) – gali pagerinti spartą bei sumažinti tinklo apkrovą laikinai išsaugodamas ir laikydamas neseniai naudotus WAP turinio resursus savo atmintinėje.

Ši infrastruktūra užtikrina tai, kad mobiliųjų terminalų vartotojai gali pasiekti įvairų interneto turinį bei taikomąsias programas. Taip pat ši infrastruktūra užtikrina tai, kad programinės įrangos gamintojai gali sukurti turinio paslaugas bei taikomąsias programas, kurios gali veikti naudojamos didelio mobiliųjų terminalų skaičiaus.WAP spartinantysis tarpinis serveris leidžia turinį ir taikomąsias programas talpinti standartiniuose pasaulinio žiniatinklio serveriuose ir programuoti naudojant jau pasiteisinusias WWW technologijas, tokias kaip CGI scenarijų rašymas, PHP ir kt.

Nors dažniausiai pasitaikantis WAP vartojimas susidės iš pasaulinio žiniatinklio serverio, WAP spartinančiojo tarpinio serverio bei WAP kliento, WAP architektūra gali gan lengvai palaikyti ir kitas konfigūracijas.

8.1.4.4 Pagalbiniai serveriai



Pagalbiniai serveriai

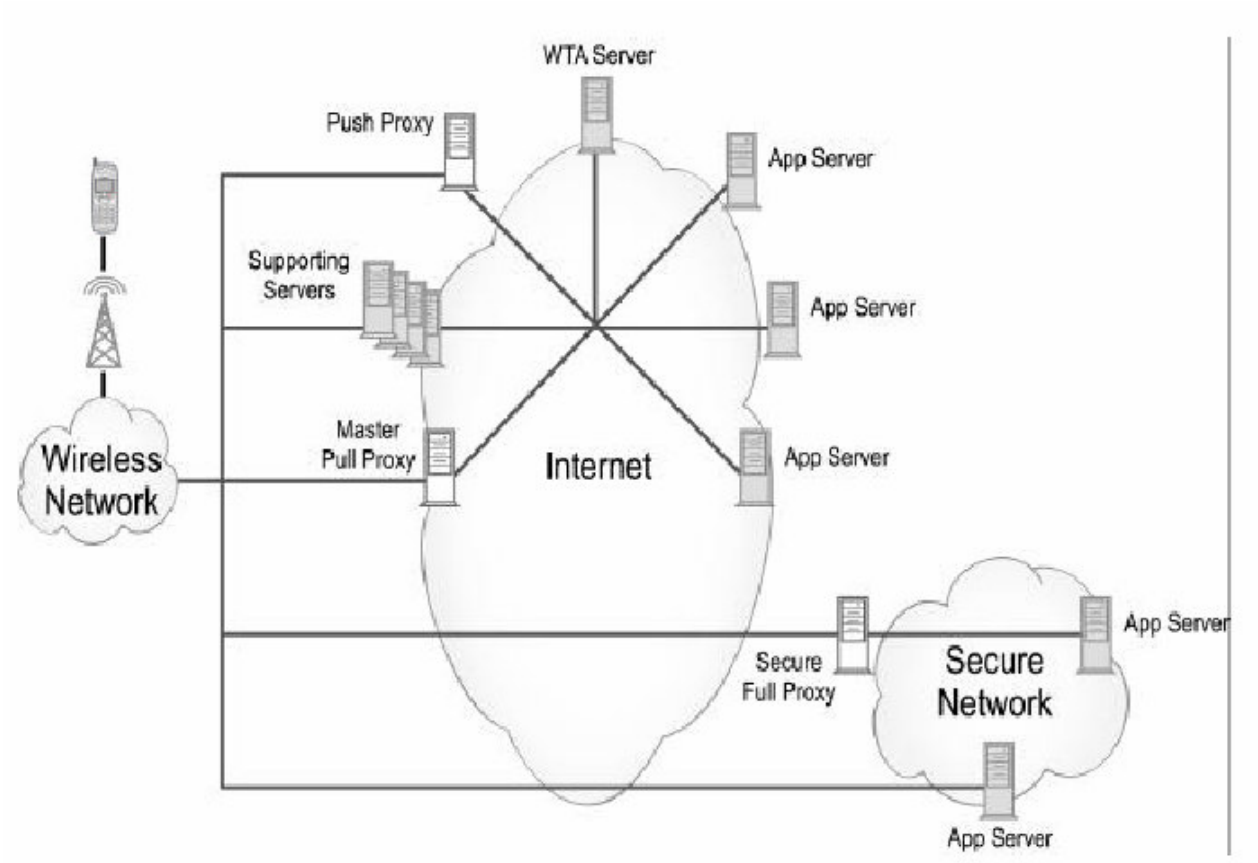
WAP architektūroje taip pat yra pagalbiniai serveriai, kurie teikia paslaugas įrenginiams, tarpiniams serveriams bei kai reikia – taikomajai programinei įrangai.Šios paslaugos dažnai atlieka specifines funkcijas, bet yra bendro naudojimo, plačiam taikomųjų programų spektrui.

Pagalbiniai serveriai kuriuos apibrėžia WAP Forum (čia pateikta tik keletas):

- PKI Portal – PKI portalas (matomas ankstesnėje iliustracijoje) [7] leidžia įrenginiams inicijuoti naujų viešo rakto (*jis*) sertifikatų sukūrimą.
- UAProf Server – šis serveris leidžia taikomosioms programoms sužinoti klientų galimybes bei asmeninius vartotojų bei jų agentų profilius.
- Provisioning Server – Šio serveriu WAP įrenginiai pilnai pasitiki, ir iš jo gauna tiekimo informaciją

8.1.4.5 WAP tinklo elementai

Čia matoma tipinė WAP tinklo schema



Pavyzdinė WAP tinklo schema

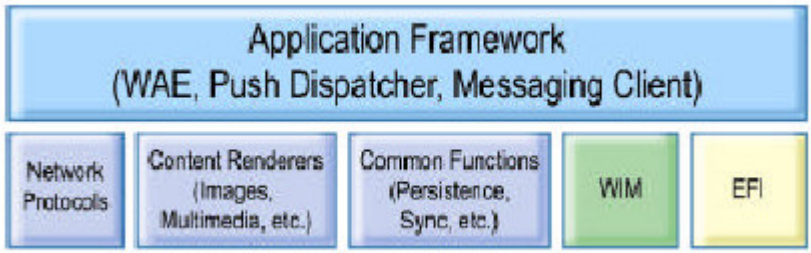
WAP klientai komunikuoja su taikomųjų programų serveriais per kelis skirtingus tarpinius serverius arba tiesiogiai. WAP klientai palaiko tarpinių serverių pasirinkimo mechanizmą kuris leidžia jiems naudoti labiausiai tinkamą tarpinį serverį duotai paslaugai, arba esant reikalui – netgi jungtis tiesiai prie paslaugos. Tarpiniai serveriai gali būti naudojami užklausos papildymui. Jie „vertėjauja“ tarp WAP ir žiniatinklio protokolų (HTTP, TCP), taip leisdami WAP klientams siųsti užklausas reikiamam serveriui.

Tarpiniai serveriai gali būti skirtingose vietose, įskaitant belaidžius nešėjus arba nepriklausomus paslaugų tiekėjus, tam kad tiekti savybių pagerinimus surištus su belaidžiu tinklu, pavyzdžiui telefonija, vietos nustatymas, nešėjo paslaugų gavimas (angl. *provisioning*), arba optimizuoti komunikavimą tarp įrenginio ir taikomųjų programų serverio (pavyzdžiui protokolų vertimas, arba slapukų (angl. *cookie*) spartinimo (angl. *caching*). Tarpiniai serveriai gali būti saugiame tinkle, tam kad tiekti saugų komunikavimo kanalą tarp belaidžio įrenginio ir saugaus tinklo.

Kai kuriais atvejais įrenginiai gali jungtis tiesiogiai prie taikomųjų programų serverių, pavyzdžiui tam kad pasiekti maksimaliai saugų tiesioginį susijungimą tarp šių dviejų taškų (kliento – serverio).

Pagalbiniai serveriai teikia pagalbines funkcijas kurios yra būtinos arba paprastai naudingos įrenginiams, tarpiniams serveriams bei taikomųjų programų serveriams. Šios funkcijos gali būti aprūpinimas (angl. *provisioning*), PKI, vartotojų agentų profiliai ir kt.

8.1.4.6 Įrenginių architektūra



WAP kliento architektūra

WAP įrenginių architektūra yra parodyta šiame paveikslėlyje. Taikomųjų programų karkasas (angl. *Application Framework*) įrenginiui teikia vykdymo aplinką WAP taikomosioms programoms. WAP taikomosios programos yra sudarytos iš žymų, scenarijaus, stiliaus lentelių bei daugialypės terpes turinio – visa tai yra perteikiama įrenginyje. WAP taikomųjų programų aplinkos WAE (angl. *WAP Application Environment*) apdorojimo modelis nurodo struktūrą, kurioje šios įvairios vykdomojo ir nevykdomojo turinio formos sąveikauja.

WAP kliento protokolai yra naudojami tiek kliento ir serverio. Toliau jie yra aprašomi detaliau. Turinio atvaizduotojai (angl. *renderers*) interpretuoja specifines turinio formas, ir jas pateikia galutiniam vartotojui, kad šis galėtu jas peržiūrėti bei atlikti norimus veiksmus. Bendros funkcijos yra nustatytos taip, kad būtų išnaudojamos taikomosios programos karkaso, įskaitant nuolatinį buvimą (angl. *persistance*) bei duomenų sinchronizavimą.

Belaidžio identifikavimo modulis WIM (angl. *Wireless Identity Module*), kaip pažymėta [7], turi savyje įrenginio tapatumą (angl. *identity*) bei kriptografines priemones abipusiam WAP įrenginių bei serverių tapatumo patvirtinimui.

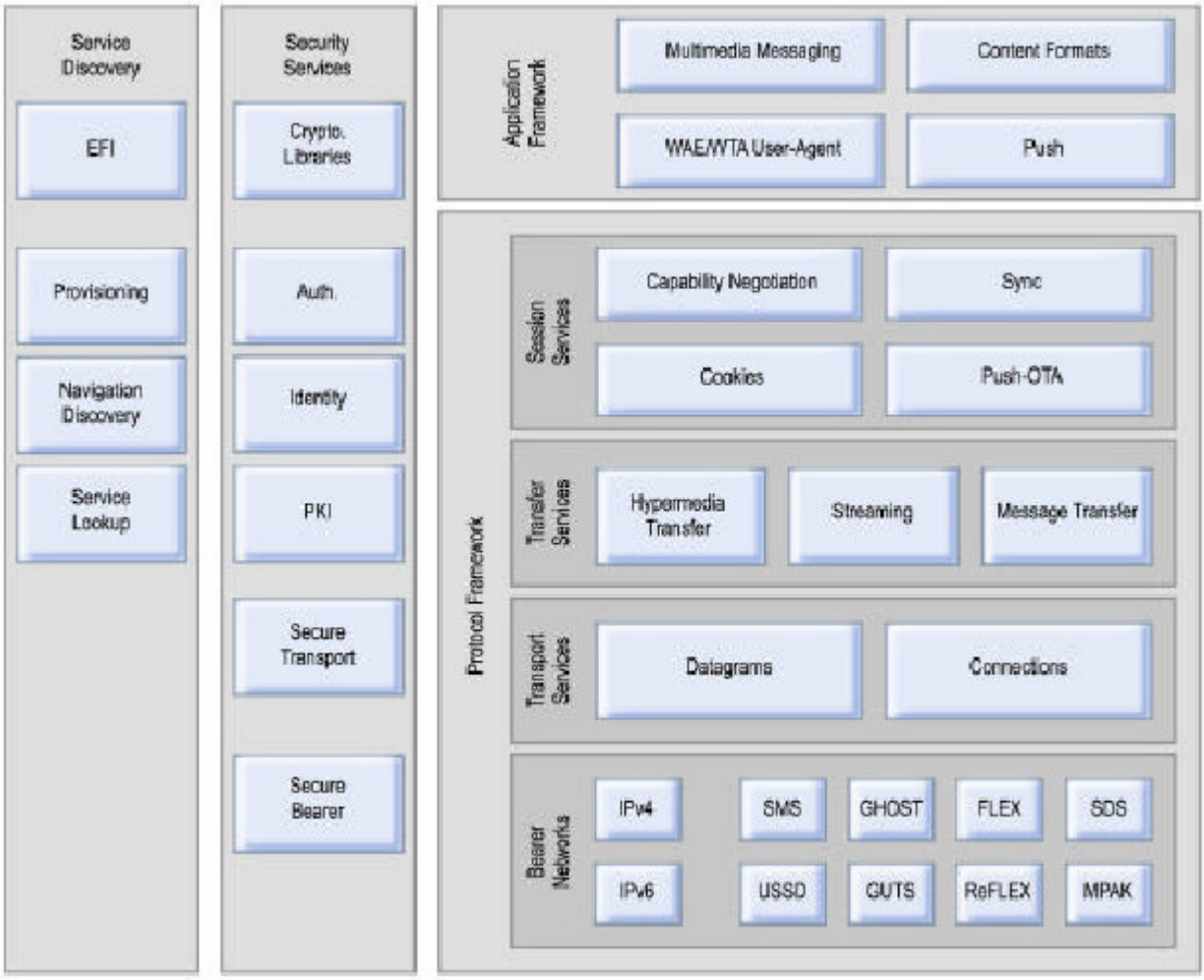
Architektūra taip pat teikia mechanizmą prieiti prie išorinių funkcijų kurios yra įmontuotos arba pridėtuos prie įrenginių naudojant išorinio funkcionalumo sąsają (angl. *External Functionality Interface - EFI*).

8.1.4.7 Saugumo modelis

WAP suteikia labai lanksčią saugumo infrastruktūrą, kuri sufokusuota tiekti susijungimo saugumą tarp WAP kliento ir serverio.

WAP gali suteikti baigtis-baigtis (angl. *end-to-end*) saugumą tarp protokolo galinių taškų. Jei naršyklė bei šaltinio serveriui reikia baigtis-baigtis saugumo, jie gali komunikuoti tiesiogiai, naudodami saugumo protokolus. Dar daugiau – WAP specifikacijose yra taikomosios programos lygio saugumo palaikymas, pvz. pasirašyto teksto (angl. *signed text*) ir pan.

8.1.5 WAP architektūros komponentai



WAP dėklo architektūra

WAP architektūra teikia keičiamos apimties bei išplečiamą taikomųjų programų kūrimo aplinką mobiliesiems komunikavimo įrenginiams. Tai yra pasiekiama naudojant sluoksnuotą protokolų dėklo architektūrą. Kiekvienas sluoksnis suteikia funkcijų ir / arba paslaugų aibę kitoms paslaugoms ir taikomosioms programoms per gerai apibrėžtų sąsajų aibę. Kiekvienas architektūros sluoksnis yra pasiekiamas aukštesniuose sluoksniuose, o taip pat paslaugoms bei taikomosioms programoms.

WAP architektūra atskiria paslaugų sąsajas nuo protokolų kurie teikia šias paslaugas. Tai padaryta specifikacijų evoliucionavimo labui, bei norint įgalinti tinkamiausio duotam kontekstui protokolo pasirinkimą. Didelė dėklo paslaugų dalis gali būti teikiamos daugiau nei vienu protokolu. Kaip pavyzdys – Hypermedia Transfer paslauga, kuri gali būti teikiama tiek HTTP [11], tiek ir WSP [7].

8.1.5.1 Nešantieji tinklai

Protokolai buvo suprojektuoti arba parinkti tam, kad veiktų per įvairias skirtingas nešančiąsias paslaugas, įskaitant trumpąsias žinutes, fiksuotus telekomunikacijų tinklus (angl. *circuit-switched*

data), duomenų perdavimo tinklus (angl. *packet-switched networks*). Nešėjai suteikia skirtingo lygio paslaugų kokybę, atsižvelgiant į pralaidumą, klaidų dažnumą ir vėlinimą. Protokolai yra sukurti tam, kad kompensuotų arba toleruotų šiuos kintančius paslaugų lygius.

Kadangi transporto paslaugų (angl. *Transport Services*) sluoksnis teikia sąsają tarp nešėjo paslaugos ir likusio WAP dėklo, transporto specifikacijos (pvz., WDP) gali išvadinti palaikomus nešėjus, bei metodus naudojamus konkrečiam protokolui veikti kiekvienu iš nešėjų. Palaikomų nešėjų sąrašas bėgant laikui keisis – plečiantis belaidžių paslaugų rinkai atsiranda vis nauji nešėjai.

8.1.5.2 Transporto paslaugos

Transporto paslaugų (angl. *Transport Services*) sluoksnis teikia pastovių paslaugų aibę aukštesnio lygio protokolams, bei susieja šias paslaugas su turimomis nešančiųjų tinklų paslaugomis. Transporto paslaugos perduoda nestruktūrizuotus duomenis per naudojamus nešančiuosius tinklus. Šios transporto paslaugos sukuria bendra abstrakciją, kuri yra bendra visiems nešėjams.

Į Transporto paslaugas įeina (bet tuo neapsiriboja):

- Datagramos – datagramų paslauga įgalina tokį duomenų transportavimą, kai nepriklausomi duomenų paketai turi savyje visą maršrutizavimui reikalingą informaciją tam, kad galėtų būti perduoti tinklu iš vieno kompiuterio į kitą nereikalaujant priklausomybės nuo ankstesnio duomenų pasikeitimo tarp šių kompiuterių bei transportavimo tinklo. UDP (angl. *User Datagram Protocol* – vartotojo datagramų protokolas) [10] ir WDP (angl. *Wireless Datagram Protocol* – bevielis datagramų protokolas) [7] yra du protokolai kurie yra naudojami tam, kad teikti datagramų transportavimo paslaugas WAP architektūroje.
- Susijungimai (angl. *connections*) – susijungimų paslauga teikia duomenų transportavimo paslaugą, kurioje transportavimas vyksta trimis aiškiai apibrėžtomis fazėmis: susijungimo įkūrimas, abipusis patikimas duomenų perdavimas, susijungimo užbaigimas. TCP (angl. *Transmission Control Protocol* – perdavimo kontrolės protokolas) [10] – yra protokolas naudotas tiekti susijungimo transportavimo paslaugas IP1 nešėjams WAP architektūroje. Tam kad „susitvarkyti“ su belaidžio tinklo charakteristikomis, TCP protokolas gali būti pritaikomas būtent šiam atvejui, žr. [20].

8.1.5.3 Perdavimo paslaugos

Perdavimo (angl. *transfer*) paslaugos įgalina struktūrizuotą informacijos perdavimą tarp tinklo elementų.

Į perdavimo paslaugas įeina (bet tuo neapsiriboja):

- Hiperdaugialypės terpės perdavimas (angl. *Hypermedia Transfer*) – hiperdaugialypės terpės perdavimo paslaugos įgalina save aprašančios hiperdaugialypės terpės resursų perdavimą. Belaidžio sesijos protokolo WSP (angl. *Wireless Session Protocol*) ir belaidžio transakcijų protokolo WTP (angl. *Wireless Transaction Protocol*) kombinacija tiekia hiperdaugialypės terpės perdavimo paslaugas tiek saugiu tiek ir nesaugiu datagramų transportavimu. Tuo tarpu HTTP (hiperteksto perdavimo protokolas) [11] tiekia hiperdaugialypės terpės perdavimo paslaugas per saugius ar nesaugius susijungimu pagrįstus transportavimu.
- Srauto perdavimas (angl. *streaming*) – srauto perdavimo paslaugos suteikia galimybę perduoti sinchroninius duomenis srautais, kas yra reikalinga norint perduoti audio arba video duomenis.
- Pranešimų perdavimas (angl. *message transfer*) – pranešimų perdavimo paslaugos suteikia galimybę asinchroninių daugialypės terpės pranešimų, tokių kaip elektroninis paštas, skubios žinutės, perdavimui. MMS įvilkimas (angl. *MMS Encapsulation*) yra protokolas naudojamas perduoti pranešimus tarp WAP įrenginių ir MMS serverių.

8.1.5.4 Sesijų paslaugos

Sesijų paslaugos skirtos bendros būsenos (angl. *shared state*) tarp tinklo elementų siunčiančių keletą tinklo užklausų ar vykdančių keletą duomenų perdavimų tuo pačiu metu, nustatymui. Pavyzdžiui „*Push*“ sesija nustato, kad WAP įrenginys gali gauti „*Push*“ pranešimus iš „*Push*“ tarpinio serverio.

Į sesijų paslaugas įeina (bet tuo neapsiriboja):

- Galimybių suderinimas (angl. *Capability Negotiation*) – WAP architektūroje yra specifikacijos skirtos aprašyti, perduoti, bei valdyti galimybių bei nustatymų informaciją apie klientą, vartotoją bei tinklo elementus (daugiau informacijos yra). Tai leidžia informacijos bei turinio, kurį gražina šaltinio serveris, pritaikymą individualiam vartotojui
- *Push*-OTA (angl. *Push Over The Air*) – ši sesijų paslauga skirta tam, kad tinklo inicijuotos transakcijos galėtų būti pristatomos belaidžiams įrenginiams, kurie ne visada gali priimti duomenis (pvz. modaliniai įrenginiai, arba įrenginiai kuriems adresas priskiriamas dinamiškai). *Push*-OTA paslauga veikia per sujungimais pagrįstas bei datagramų transportavimo paslaugas [*7*].
- Sync – ši sinchronizacijos paslauga skirta sinchronizuoti replikuojamus duomenis.

- Slapukai (angl. *cookies*) – slapukų paslauga leidžia taikomosioms programoms nustatyti būsenas cliente arba tarpiniame serveryje, kurios išlieka daugiau nei vieną hiperdaugialypės terpės perdavimo transakciją. Daugiau informacijos šia tema yra [4].

8.1.5.5 Taikomųjų programų karkasas

Taikomųjų programų karkasas suteikia bendros paskirties programų aplinką pagrįstą žiniatinklio (WWW), interneto ir mobiliosios telefonijos technologijomis. Pirminis taikomųjų programų karkaso tikslas yra sukurti plačiai veikiančią aplinką, kuri leis operatoriams ir paslaugų tiekėjams kurti taikomąsias programas ir paslaugas kurios gali pasiekti platų skirtingų bevielių platformų spektrą efektyviu ir naudingu būdu.

Į taikomųjų programų karkasą įeina (bet tuo neapsiriboja):

- WAE/WTA vartotojo agentas – WAE yra mikro-naršyklės aplinka kuri turi arba įgalina žymų kalbas (įskaitant WML ir XHTML), scenarijų rašymą, stilių lentelių kalbas, telefonijos paslaugas bei programavimo sąsajas. Visi šie dalykai yra optimizuoti mažų gabaritų belaidžiams mobiliems terminalams. Daugiau informacijos galima rasti [8].
- *Push* – ši paslauga suteikia bendrą mechanizmą tinklui inicijuoti duomenų siuntimą į WAP įrenginiuose esančias taikomąsias programas. Daugiau informacijos galima rasti [7].
- Daugialypės terpės pranešimai MMS (angl. *Multimedia Messaging Service*) – įgalina daugialypės terpės pranešimų, tokių kaip elektroninis paštas, skubios žinutės, perdavimą į WAP įrenginius, bei atvaizdavimą juose.
- Turinio formatai (angl. *content formats*) – taikomųjų programų karkasas palaiko aibę gerai apibrėžtų domenų formatų, pvz. spalvoti vaizdai, audio, video, animacija, telefonų knygos įrašai, kalendoriaus informacija.

8.1.5.6 Saugumo paslaugos

Saugumo savybės yra viena iš pagrindinių WAP architektūros dalių, ir jos paslaugos gali būti randamos daugelyje architektūros sluoksnių. Apibendrinus, yra teikiamos šios saugumo paslaugos:

- Privatumas (angl. *privacy*) – paslaugos, skirtos užtikrinti kad komunikavimas yra privatus, t.y. jis negali būti suprastas jokių kitų tarpinių šalių, kurios galbūt jį priėmė.
- Autentiškumo patvirtinimas (angl. *authentication*) – paslaugos, leidžiančios nustatyti komunikuojančių šalių autentiškumą.
- Originalumas (angl. *integrity*) – paslaugos, užtikrinančios kad komunikavimas nėra pakeistas, modifikuotas, arba pažeistas.

- Pripažinimas (angl. *non-repudiation*) – paslaugos, užtikrinančios jog komunikuojančios pusės negalės paneigti komunikavimo fakto.
- Saugumo paslaugos yra įvairiuose WAP architektūros sluoksniuose. Kai kurie specifiniai saugumo paslaugų pavyzdžiai:
    - Kriptografinės bibliotekos: šio taikomųjų programų karkaso sluoksnio bibliotekos teikia duomenų originalumo ir pripažinimo tikslams reikalingas parašų paslaugas. Daugiau informacijos šia tema yra [8].
    - Autentiškumo patvirtinimas – saugumo paslaugos teikia įvairius mechanizmus skirtus kliento bei serverio autentiškumo patvirtinimui. Saugumo paslaugų sluoksnyje klientų autentiškumo patvirtinimui tarpiniams serveriams bei taikomųjų programų serveriams gali būti naudojamas HTTP Kliento autentiškumo patvirtinimas (angl. *HTTP Client Authentication*) [14]. Tuo tarpu transportavimo paslaugų sluoksnyje, WTLS ir TLS pasisveikinimai (angl. *handshakes*) gali būti naudojami klientų bei serverių autentiškumo nustatymui.
    - Tapatumas – WIM teikia funkcijas, reikalingas išsaugoti ir apdoroti informaciją reikalingą vartotojo tapatumo bei autentiškumo nustatymui.
    - PKI – saugumo paslaugų aibė, leidžianti viešojo rakto kriptografijos bei sertifikatų [7] panaudojimą bei valdymą.
    - Saugus transportavimas – transportavimo paslaugų sluoksnio protokolai yra apibrėžti, tam kad įgalinti saugų perdavimą naudojant datagramas bei sujungimus. WTLS yra skirtas saugiam perdavimui naudojant datagramas, o TLS yra skirtas saugiam duomenų transportavimui naudojant sujungimus (pvz. TCP). Daugiau informacijos šia tema yra [7].
    - Saugus nešėjas – kai kurie nešantieji tinklai suteikia papildomą saugumą jau nešančiųjų tinklų lygyje. Pavyzdžiui IP tinklai (ypač kalbant apie IPv6) suteikia saugumą nešančiojo tinklo lygyje naudojant IPSec [15].

8.1.5.7 Paslaugų atradimas

Paslaugų atradimas yra viena iš pagrindinių WAP architektūros dalių, ir jos paslaugos gali būti randamos daugelyje architektūros sluoksnių.

Pateiksime keletą specifinių paslaugų atradimo pavyzdžių:

- EFI (angl. *External Functionality Interface*) – išorinio funkcionalumo sąsaja, leidžianti atrasti išorines funkcijas ir paslaugas kurios yra prieinamos įrenginyje.
- Tiekimas (angl. *Provisioning*) – tiekimo paslaugos leidžia įrenginiui reikti parametrus, reikalingus norint prisijungti prie tinklo paslaugų.).

- Navigacinis atradimas (angl. *Navigation Discovery*) – navigacinio atradimo paslauga leidžia įrenginiui atrasti naujas tinklo paslaugas (pavyzdžiui „*Pull*“ tarpinius serverius) navigacijos eigoje, pavyzdžiui gaunant duomenis iš hiperdaugialypės terpės serverio. WAP transportavimo lygio baigtis-baigtis saugumo (angl. *WAP Transport-Level End-to-End Security*) specifikacija TransportE2ESec] nurodo vieną navigacinio atradimo protokolą.
- Paslaugų atradimas – paslaugų atradimo paslauga skirta paslaugos parametrams atrasti, ieškant kataloge pagal vartą. To pavyzdys – srities vardų sistema DNS (angl. *Domain Name System*) [16].

8.1.5.8 Kitos paslaugos bei taikomosios programos

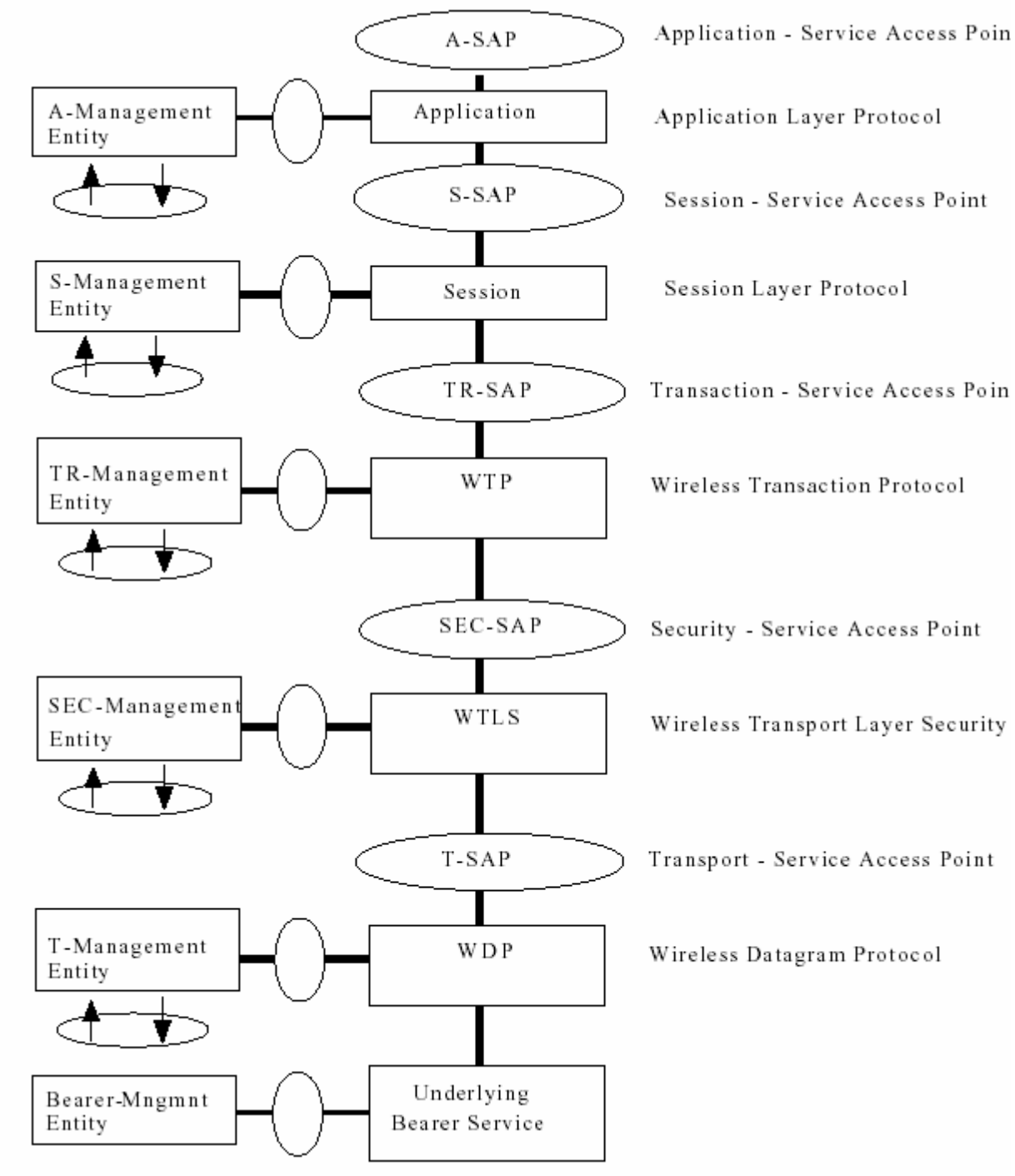
WAP sluoksninė architektūra leidžia kitoms paslaugoms bei programoms išnaudoti WAP dėklo savybes, naudojant gerai apibrėžtų sąsajų aibę. Išorinės taikomosios programos gali prieiti prie įvairių paslaugų tiesiogiai. WAP sluoksninė architektūra yra sudaryta iš išplečiamos protokolų aibės. Tai leidžia WAP dėklą panaudoti taikomosioms programoms ir paslaugoms kurios einamuoju momentu nėra specifikuotos WAP, tačiau manoma kad yra naudingos belaidžio tinklo rinkai. Tokios taikomosios programos bei paslaugos gali gauti naudą įdėdami naujus protokolus, ar tam tikras protokolų galimybes. Pavyzdžiui tokie taikymai kaip elektroninis paštas, kalendorius, telefonų knyga, bloknotas, elektroninė komercija, arba paslaugos kaip baltieji ir geltonieji puslapiai, gali būti sukurti naudojant WAP protokolus.

8.1.6 Belaidžių datagamų protokolo WDP architektūros apžvalga

WDP (angl. *Wireless Datagram Protocol*) protokolas dirba naudodamas galinčias perduoti duomenis nešančiųjų tinklų paslaugas, palaikomas ne vieno tipo tinklo. WDP suteikia pastovią paslaugą „aukštesniems“ WAP protokolams (pvz. saugumo, perdavimo ir sesijų) ir nepastebimai (angl. *transparently*) komunikuoja naudodamas vieną iš galimų nešėjų paslaugų.

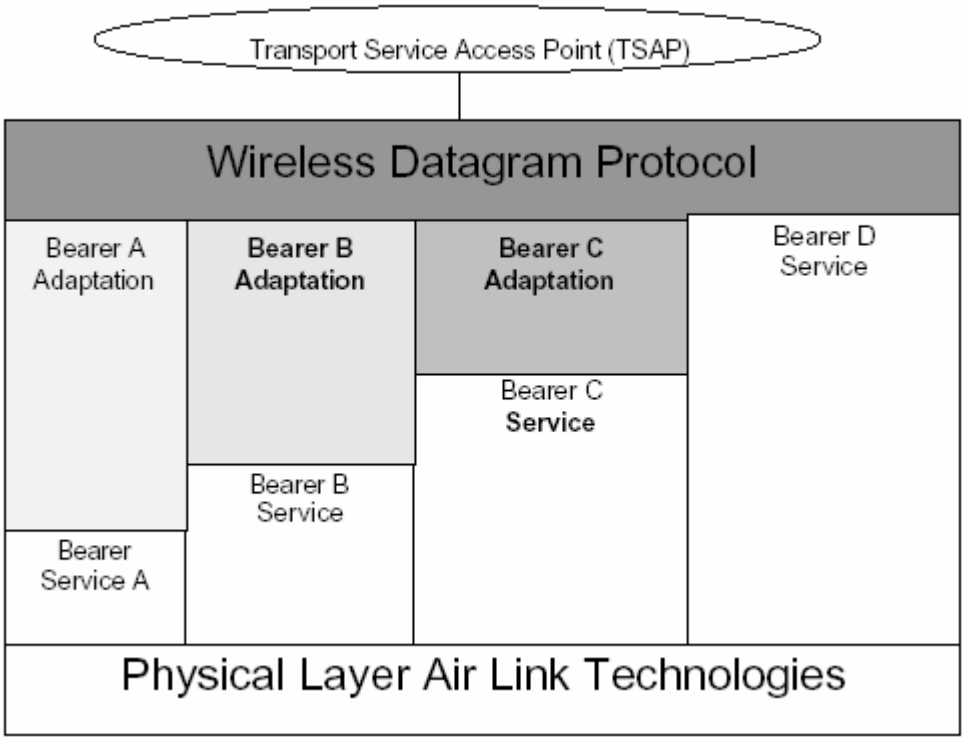
8.1.6.1 Standarto modelis

Šioje schemoje matoma belaidžių datagramų protokolo vieta WAP architektūriniame modelyje



WAP standarto architektūra

WDP paslaugos yra programų adresavimas pagal prievadų numerius, laisvai pasirenkama segmentacija ir surinkimas, galimas klaidų aptikimas. Paslaugos leidžia programoms veikti vienodai, neatsižvelgiant į skirtingas galimas nešėjo paslaugas.



Belaidžių datagramų protokolo architektūra

Belaidžių datagramų protokolo architektūros modelis buvo pateiktas šioje schemoje. WDP teikia pastovias paslaugas transporto paslaugų prisijungimo taške (angl. *Transport Service Access Point*) aukštesnio lygio WAP protokolui. Šis paslaugos pastovumas leidžia taikomosioms programoms veikti vienodai, neatsižvelgiant į skirtingas galimas nešėjo paslaugas. Skirtingi nešančių paslaugų aukščiai parodyti schemoje, rodo skirtumus tarp įvairių nešėjų siūlomų paslaugų ir tuo pačiu – skirtumus būtinus WDP protokolo veikime tam, kad būtų galima dirbti su įvairiais nešėjais išlaikant tas pačias transporto paslaugų prisijungimo taško (angl. *Transport Service Access Point*) paslaugas.

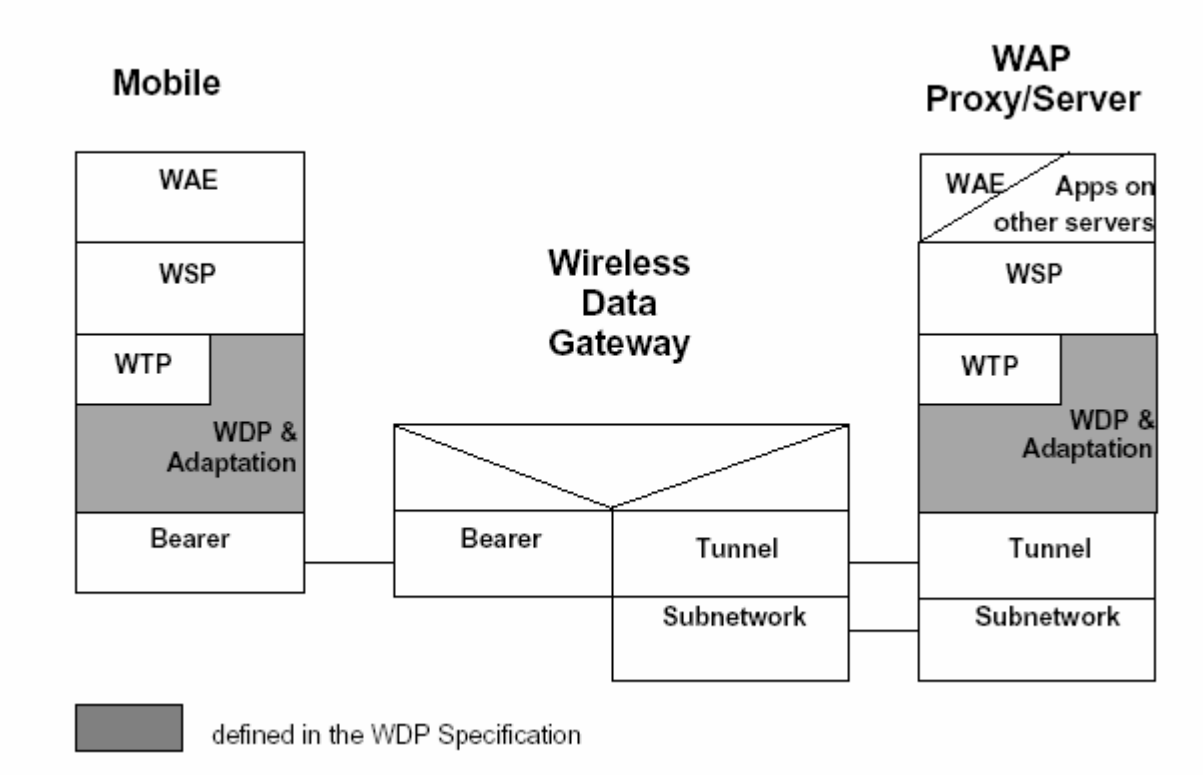
WDP gali būti susietas su įvairiais nešėjais, turinčiais skirtingas charakteristikas. Tam kad optimizuoti protokolą atsižvelgiant į atminties vartojimą ir radijo signalo efektyvumą, protokolo sparta bei kitos charakteristikos gali skirtis naudojant skirtingus nešėjus. Tačiau WDP paslaugos baziniai elementai bus tie patys, teikiantys pastovias sąsajas aukštesniems sluoksniams.

8.1.6.2 Bendras WDP protokolo aprašymas

WDP sluoksnis veikia „virš“ duomenis perduoti galinčių nešėjo paslaugų, kurias palaiko įvairūs tinklų tipai. Kaip bendra datagramų perdavimo paslauga, WDP siūlo pastovias paslaugas „aukštesnio“ lygio WAP protokolams (saugumo, saugumo, perdavimo ir sesijų) komunikuoti abstrakčiai naudojant vieno iš galimų nešėjų paslaugas. WDP palaiko keletą vienu metu esančių sujungimų iš „aukštesnio“ sluoksnio per vieną „žemiau“ esančią nešėjo paslaugą. Sąsajos (angl. *port*) numeriai identifikuoja „aukštesnio“ sluoksnio esybes. Tai gali būti tiek kitų protokolų sluoksniai, tokie kaip belaidžių

transakcijų protokolas WTP (angl. *Wireless Transaction Protocol*) arba belaidžių sesijų protokolas WSP (angl. *Wireless Session Protocol*), tiek ir taikomosios programos, pavyzdžiui elektroninio pašto. Išnaudodamas „žemiau“ esančių nešėjų elementus, WDP gali būti sukurtas palaikyti keletą nešėjų, ir tuo pačiu būti opimiziotas efektyviam veikimui naudojant ribotus mobiliojo įrenginio resursus.

Sekanti schema demonstruoja bendrą WAP protokolo architektūros modelį, ir tai, kaip WDP egzistuoja šioje architektūroje.



WAP protokolo architektūros modelis ir WDP

8.1.6.2.1 WDP valdymo esybė

WDP valdymo esybė yra naudojama kaip sąsaja tarp WDP sluoksnio bei įrenginio aplinkos. WDP valdymo esybė tiekia informaciją WDP sluoksniui apie pasikeitimus įrenginio aplinkoje, kurie gali įtakoti teisingą WDP veikimą.

WDP protokolas yra sukurtas remiantis prielaida kad naudojama aplinka gali perduoti ir gauti duomenis. Kaip prielaidos pavyzdį, galime pateikti šias bazines galimybes, kurias turi tiekti mobilusis įrenginys:

- mobilus įrenginys yra naudojamos nešančiosios paslaugos veikimo zonoje
- mobilus įrenginys turi pakankamai energijos bei yra įjungtas
- mobiliajame įrenginyje yra pakankamai resursų (atminties bei perdirbimo galimybių) kuriais gali naudotis WDP
- WDP protokolas yra nustatytas teisingai
- vartotojas nori siųsti / gauti duomenis

WDP valdymo esybė stebi šių paslaugų / galimybių statusą mobiliojo įrenginio aplinkoje, ir informuoja WDP sluoksnį jei viena ar daugiau iš tariamų paslaugų yra nepasiekiama. Pavyzdžiui jei mobilusis įrenginys apleido nešančiosios paslaugos veikimo zoną, nešėjo valdymo esybė (angl. *Bearer Management Entity*) turėtų pranešti tai WDP valdymo esybei kad siuntimas / gavimas baigėsi, ir nešėjas neprieinamas. Savo ruožtu WDP valdumo esybė praneštų WDP sluoksniui uždaryti visus aktyvius sujungimus naudojančius tą konkretų nešėją. Kiti atvejai, tokie kaip mobiliojo įrenginio energijos baterijos išsekimas, būtų taip pat panašiai valdomi WDP valdymo esybės. Be mobiliojo įrenginio aplinkos būsenų stebėjimo, WDM valdymo esybė gali būti panaudojama kaip sąsaja skirta vartotojui nustatinėti įvairius konfigūracijos parametrus naudojamus WDP, pavyzdžiui įrenginio adreso nustatymas. Taip pat ji gali būti naudojama tokių vartotojui prieinamų funkcijų kaip „uždaryti visus sujungimus“ sukūrimui. Bendru atveju WDP valdymo esybė yra atsakinga už šios dalykus: darbo pradžia (angl. *initialization*), konfigūracija, dinaminis konfigūracijos keitimas, resursai (nes jie siejasi su WDP sluoksniu). Kadangi WDP valdymo esybė turi sąveikauti su įvairiais įrenginio komponentais, kurie paprastai priklauso nuo gamintojo, laikoma kad WDP valdymo esybės projektavimas ir įgyvendinimas nepapuola į WDP specifikacijos sritį, ir yra laikoma įgyvendinimo problema.

8.1.6.2.2 WDP datagramų klaidų apdorojimas

WDP datagramų klaidų apdorojimas reikalingas tada kai datagramos yra siunčiamos iš vieno WDP tiekėjo kitam. Pavyzdžiui belaidžių duomenų vartai (angl. *Wireless Data Gateway*) negali persiųsti duomenų WAP vartams, arba taikomoji programa nelaukia duomenų paskirties prievade (angl. *port*), arba gavėjas neturi pakankamai vietos buferyje kad priimti didelį pranešimą. Belaidis valdymo pranešimų protokolas WCMP (angl. *Wireless Control Message Protocol*) teikia efektyvų klaidų apdorojimo mechanizmą skirtą WDP, ko pasekoje WAP protokolo bei taikomųjų programų našumas pagerėja. Taigi WCMP protokolas TURĖTŲ būti įdiegtas.

Tikimasi kad WDP nepraleis klaidingų / sugadintų pranešimų. Jei tam tikras nešėjas nepalaiko šios paslaugoms, tada WDB nešėjo pritaikymas (angl. *adaptation*) PRIVALO teikti apkrovos apsaugą.

8.1.6.2.3 Saugumo svarstymas

WDP nėra autentiškumo nustatymo mechanizmų.

8.1.6.3 WDP pritaikymas

Yra tam tikra minimali privalomų įdiegti WDP savybių aibė, tam kad užtikrinti skirtingų gamintojų produkcijos teisingą tarpusavio veikimą. WDP protokolas veikia naudodamas įvairių nešėjų paslaugas. Kiekvieno nešėjo paslauga kuriai yra nurodytas WDP palaiko datagramų paslaugą. Būtent šią datagramų paslaugą WDP ir naudoja tam kad palaikyti abstrakčių paslaugų pagrindą, aprašytą

specifikacijoje. Nešėjų paslaugoms palaikančioms IP, WDP protokolas TURI būti UDP. Nešėjų paslaugoms nepalaikančioms IP, turi būti naudojamas WDP protokolas nurodytas specifikacijoje.

8.1.6.3.1 WDP adaptacijos sluoksnio segmentacija ir surinkimas

Pristatant naują tinklo nešėjo paslaugą, reikia apgalvoti galimą segmentacijos ir surinkimo SAR (angl. *Segmentation & Re-Assembly*) funkcionalumo įtraukimą į šio nešėjo paslaugos adaptacinį sluoksnį.

Svarstant ar reikia įtraukti SAR funkcionalumą naujai nešėjo paslaugai reikėtų įvertinti šiuos kriterijus:

- taikomosios programos (arba aukštesni komunikavimo sluoksniai) kurie naudos nešėjo paslaugas: įvertinti ar nauja nešėjo paslauga galės susidoroti su šių programų ir sluoksnių sudaromomis tipinėmis apkrovomis (pavyzdžiui naudojant X.509 sertifikatus su WTLS, tipinis sesijos užmezgimo pranešimas užima iki 1500 baitų)
- maksimalus nešančiosios paslaugos perduodavo vieneto dydis - MTU (angl. *Maximum Transfer Unit*)

Jei taikymo tipinė apkrova yra didesnė nei nešėjo MTU, SAR palaikymas turėtų būti įtraukiamas į naujo nešėjo paslaugos specifikaciją. Kai nešėjui nurodomas SAR, nešėjo tinklas turi bent jau galėti atpažinti ir ignoruoti segmentuotus pranešimus, jei jis negali jų priimti ir surinkti.

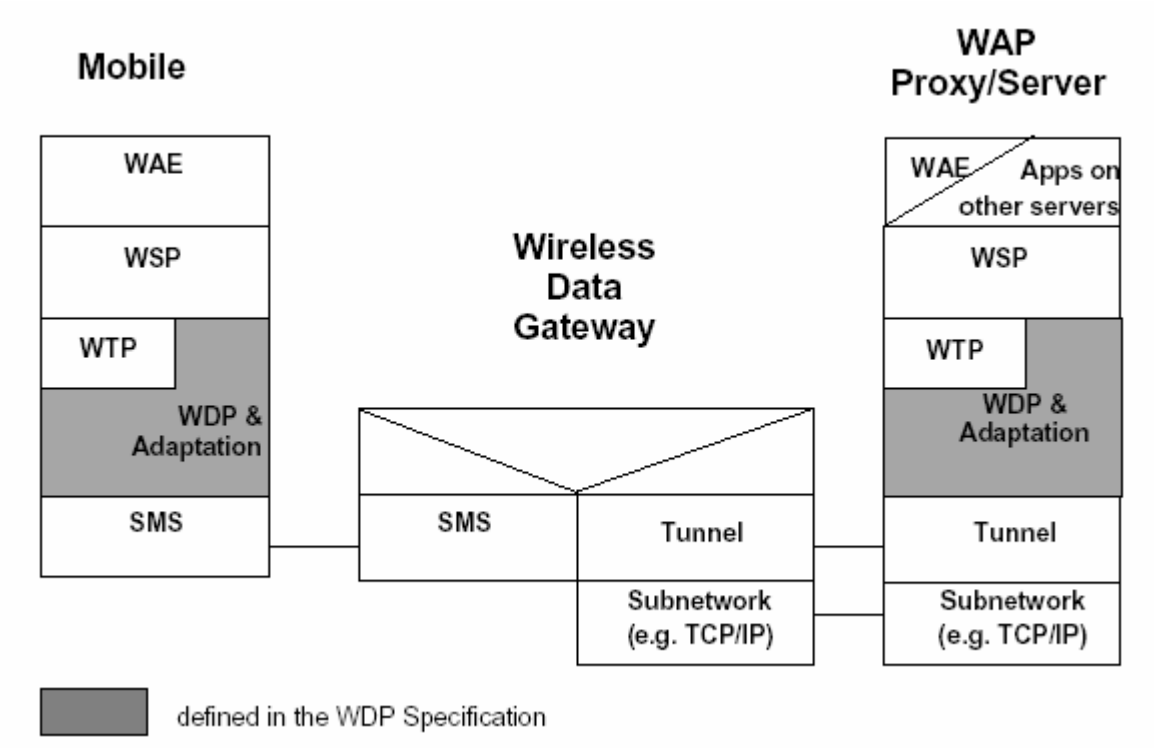
Tačiau SAR palaikymas nešėjo paslaugoje dar negarantuoja kad taikomajai programai kad duomenų transportavimas vartotojui bus savalaikis. Taigi programos turi žinoti vartotojui nebus priimtina jei dideli duomenų kiekiai bus siunčiami tam tikromis nešėjų paslaugomis.

8.1.6.4 WDP profilio priklausomybė nuo nešėjo

Sekančios schemos iliustruos WDP veikimo, tarp mobiliojo įrenginio ir serverio, protokolo profilius naudojant tam skirtingus radijo dažnius ir jiems skirtas technologijas nešėjas.

8.1.6.4.1 WDP per GSM

8.1.6.4.1.1 GSM SMS Profilis



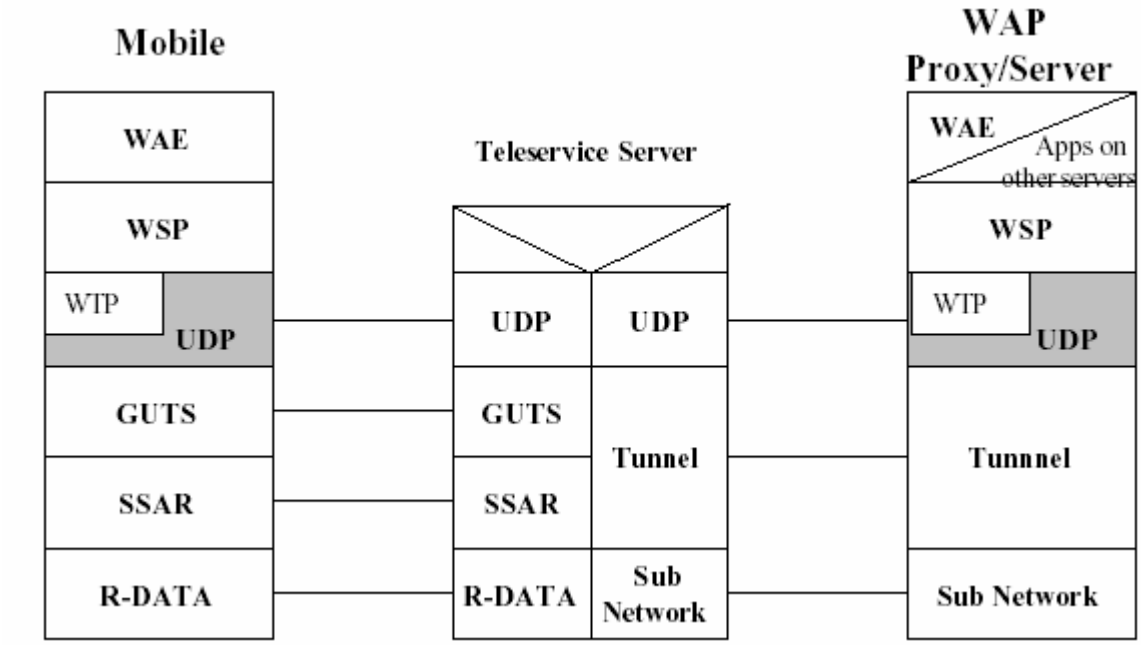
WDP per GSM SMS

Schema rodo protokolo profilį WDP sluoksniui, kai dirbama naudojant SMS nešėjo paslaugą.

8.1.6.4.2 WDP per ANSI-136

8.1.6.4.2.1 ANSI-136 R-Data profilis naudojant GUTS

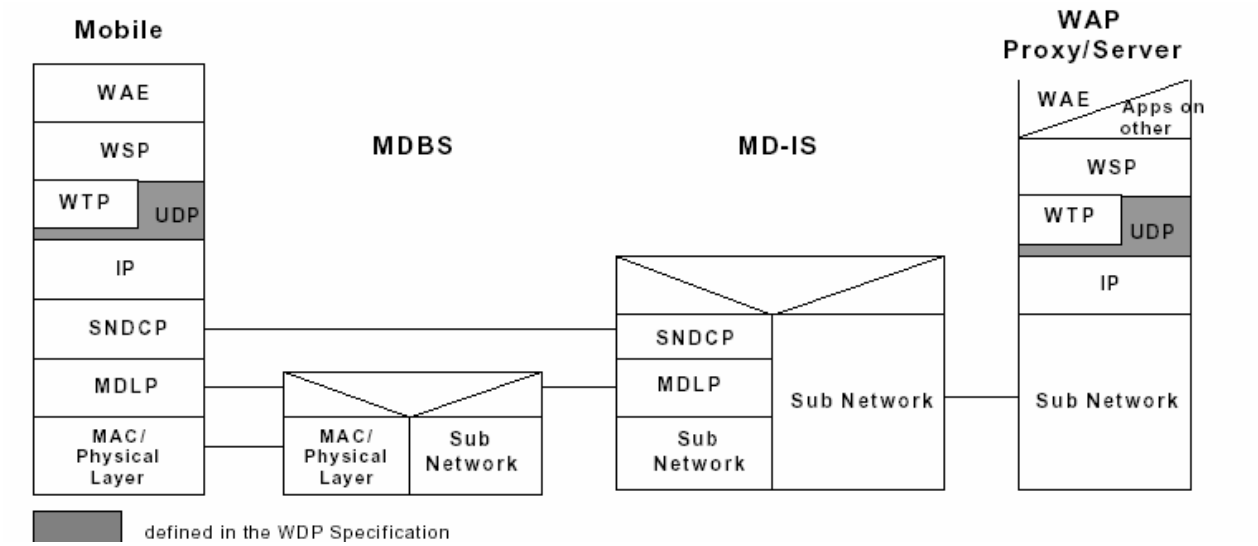
WDP sluoksnis veikia „virš“ duomenis galinčio perduoti nešėjo paslaugų, palaikomų ANSI-136.



WDP per ANSI-136 R-DATA naudojant GUTS

Schema iliustruoja protokolo profilį WDP sluoksniui, kai dirbama „virš“ ANSI-136 GUTS ir R-DATA nešėjo paslaugų. Efektyvumo sumetimais, WDP gali būti tiesiogiai palaigomas GUTS. Šiam tikslui būtų reikalingas GUTS protokolo diskriminatorius. ANSI-136 telepaslaugų serverio (angl. *Teleservice Server*) sąsajos protokolas yra priklausomas nuo potinklio (angl. *Sub Network*) ir nėra specifikuotas WAP specifikacijose.

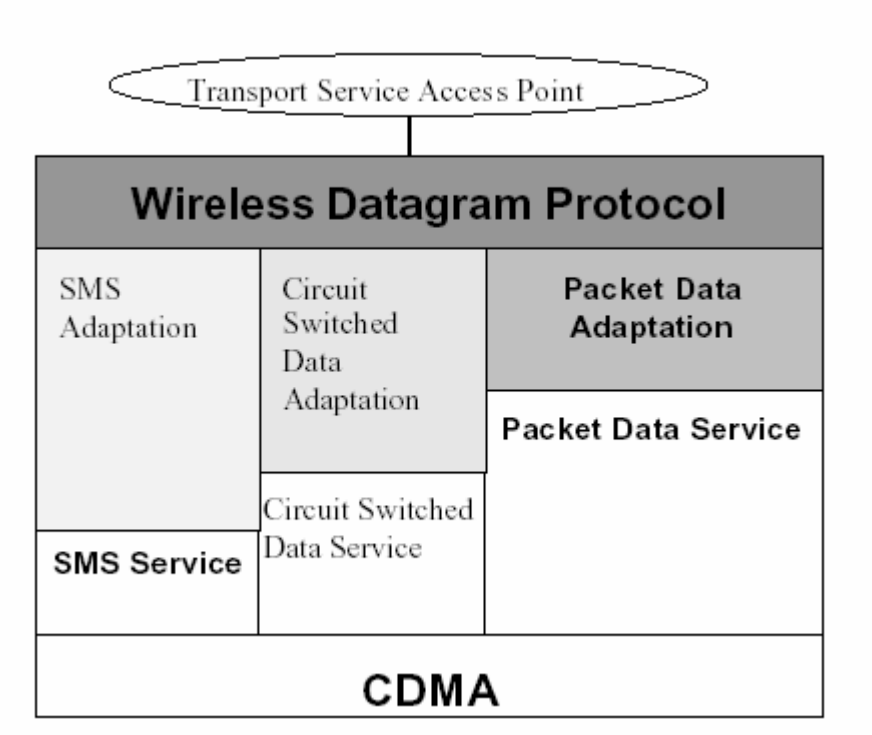
8.1.6.4.3 WDP per CDPD



WDP per CDPD

Schema iliustruoja protokolo profilį WDP sluoksniui, kai dirbama „virš“ CDPD nešėjo paslaugų. CDPD palaiko IP mobiliems, tačiau datagramų paslaugas teiks UDP/IP.

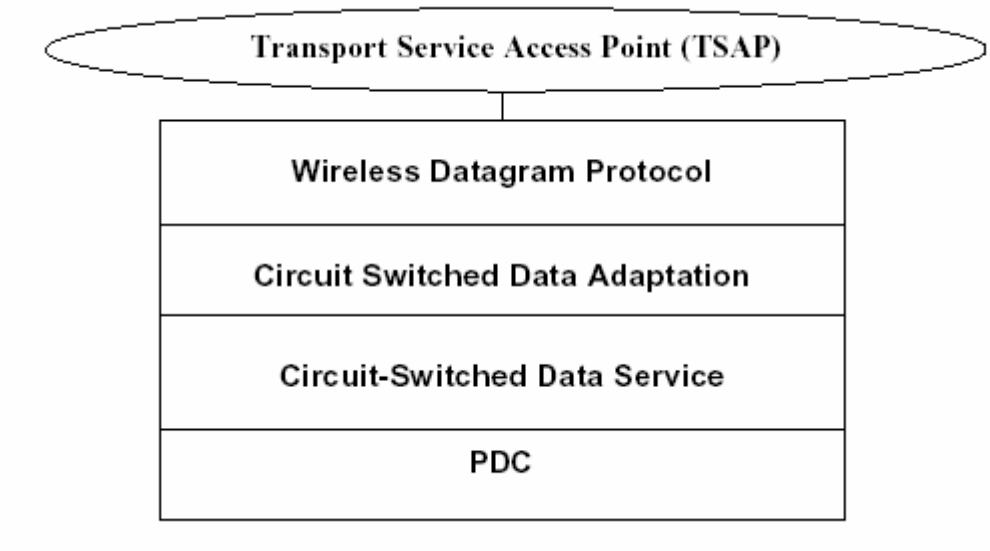
8.1.6.4.4 WDP per CDMA



WDP per CDMA nešėjo paslaugas

WDP sluoksnis dirba „virš“ duomenis galinčio perduoti nešėjo paslaugų, palaikomų CDMĄ. Ši iliustracija parodo CDMA nešėjo paslaugas, nurodytas specifikacijoje.

8.1.6.4.5 WDP per PDC



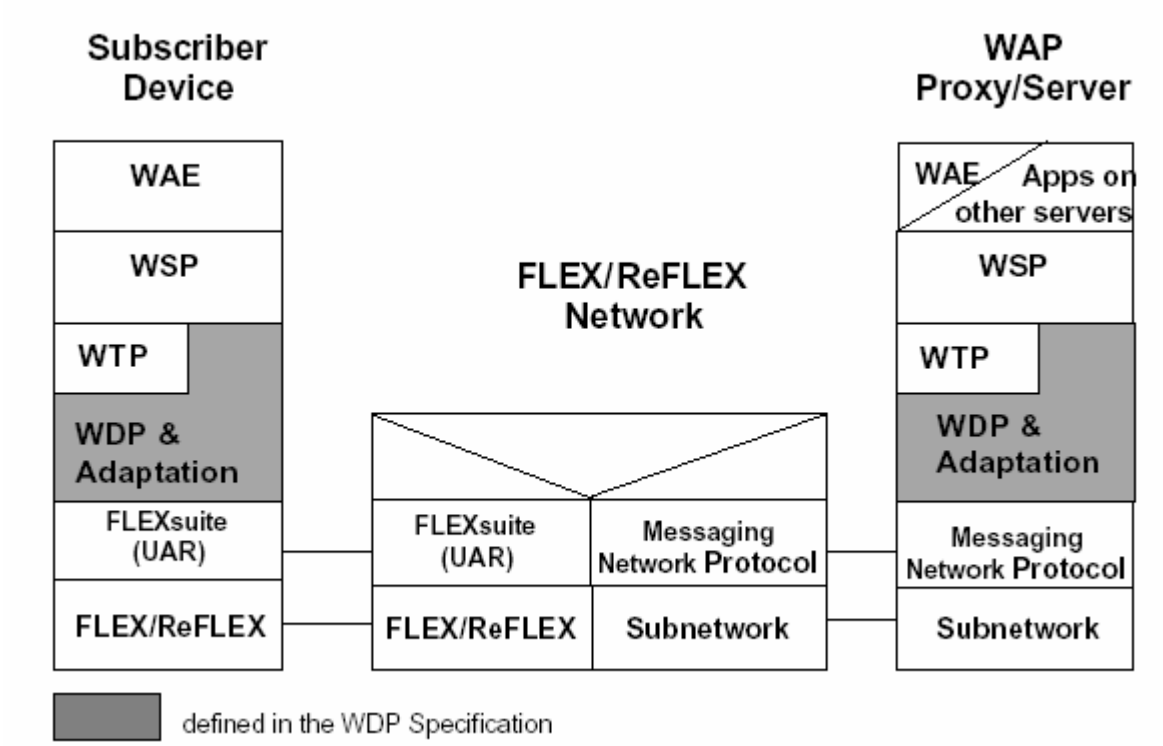
WDP per PDC nešėjo paslaugas

PDC yra skaitmeninis korinis tinklas kuriam eterio sąsaja yra apibrėžta standarto RCR STD-27 o tinklo vidinių taškų sąsajos apibrėžtos standarto TTC JJ-70.10. WDP sluoksnis veikia virš duomenis galinčio perduoti nešėjo paslaugų palaikomų PDC. Ši iliustracija parodo PDC nešėjo paslaugas, nurodytas specifikacijoje. PDC tiekia CSDS (angl. *Circuit Switched Data Service*)

8.1.6.4.6 WDP profiliai per iDEN

Nešėjas iDEN tiekia tris duomenų paslaugas, trumpų pranešimų paslaugas (angl. *Short Message Service*), fiksuotus telekomunikacijų (angl. *Circuit Switched*) bei iDEN paketinius duomenis (angl. *Packet Data*). Tiek fiksuotos telekomunikacijų, tiek paketinių duomenų paslaugos suteikia IP prisijungimą mobiliajam įrenginiui. Tačiau datagramų protokolas naudojamas iDEN duomenų nešėjo paslaugose yra UDP.

8.1.6.4.7 WDP per FLEXTM ir ReFLEXTM



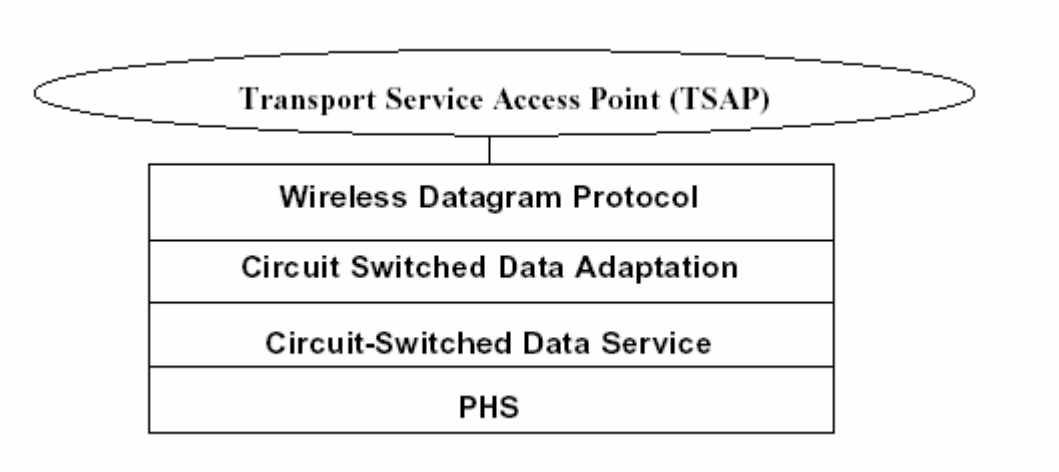
WDP per FLEX bei ReFLEX

Schema iliustruoja protokolo profilį WDP sluoksniui, kai dirbama „virš“ FLEX ir ReFLEX pranešimų protokolų. Norint kad įvyktų komunikavimas tarp FLEX/ReFLEX tinklo ir abonemento įrenginio, WDP paketai TURI būti perduoti naudojant FLEXsuite™ vienodą adresavimo ir maršrutizavimo protokolą UAR (angl. *Uniform Addressing and Routing*). Tinklo protokolas naudojamas komunikuoti tarp WAP tarpinių serverių / serverių ir FLEX/ReFLEX tinklų yra suderinamas tarp kiekvieno WAP tarpinio serverio / serverio ir FLEX/ReFLEX tinklo, ir čia nėra specifikuojamas. UAR protokolas GALI būti naudojamas tarp FLEX/ReFLEX tinklo ir WAP tarpinio serverio / serverio. Taip pat WAP tarpinis serveris / serveris gali nutraukti WAE arba tarnauti kaip tarpinis serveris kitoms taikomosioms programoms internete, arba kituose tinkluose. ReFLEX palaiko pranešimų segmentavimą. Reikia pastebėti jog tinklų operatoriai kartais apriboja priimamų pranešimų dydi. Todėl tikimasi kad tinklo operatoriai nenustatys maksimalaus pranešimo dydžio mažesnio, nei minimalus WTP leidžiamas – tokiu būdu segmentavimas bei surinkimas nebus reikalingas WDP.

8.1.6.4.8 WDP per PHS

PHS yra skaitmeninis belaidis tinklas, kurio eterio sąsaja yra aprašyta standarto RCR STD-28, o korio stotis – skaitmeninė tinklo sąsaja standartų TTC JT-Q931-b, TTC JT-Q932-a ir TTC JT-Q921-b.

WDP sluoksnis dirba „virš“ duomenis galinčio perduoti nešėjo paslaugų, palaikomų PHS. Ši iliustracija parodo PHS nešėjo paslaugas, nurodytas specifikacijoje. PHS tiekia CSDS (angl. *Circuit Switched Data Service*)



WDP per PHS nešėjo paslaugas

8.1.6.5 Elementų tarpsluoksninis komunikavimas

8.1.6.5.1 Paslaugų primityvų notacija

Komunikavimas tarp sluoksnių ir tarp esybių transportavimo sluoksnyje yra pasiekiamas paslaugų primityvų būdu. Paslaugų primityvai abstrakčiu būdu rodo loginius informacijos bei valdymo mainus tarp transporto ir gretimų sluoksnių. Jie nenurodo ir neapriboja įgyvendinimo. Paslaugų primityvai susideda iš komandų ir jų atitinkamų atsakymų, susietų su paslaugomis užklaustomis iš kito sluoksnio. Bendra primityvo sintaksė yra sekanti:

X – Bendrinis vardas. Tipas (Parametrai)

Čia X žymi sluoksnį suteikiantį paslaugą. Pavyzdžiui X reikšmė „T“ būtų skirta transportavimo sluoksniui (angl. *Transport Layer*).

Paslaugų primityvai nėra tokie patys kaip programų kūrimo sąsajos API (angl. *Application Programming Interface*), ir nėra skirtos nurodyti API kūrimo metodams. Paslaugų primityvai yra abstraktus būdas atvaizduoti protokolo lygio paslaugas teikiamas aukštesniajam sluoksniui. Šios koncepcijos susiejimas su tikru API yra įgyvendinimo problema, ir nėra šios specifikacijos dalis.

8.1.6.5.2 Paslaugų primityvų tipai

Primityvų tipai yra:

Request (.Req) (užklausa)

Užklausos primityvo tipas yra naudojamas kai aukštesnis sluoksnis prašo paslaugos iš žemesnio sluoksnio.

Indication (.Ind) (nurodymas)

Nurodymo primityvo tipas yra naudojamas teikiančio paslauga sluoksnio, norint informuoti sekantį aukštesnį sluoksnį apie to paties sluoksnio veiklą susijusią su užklausos primityvo tipu.

Response (.Res) (atsakymas)

Nurodymo primityvo tipas yra naudojamas sluoksnio norint pranešti gavėjui iš žemesnio sluoksnio, apie nurodymo primityvo tipą.

Confirm (.Cnf) (patvirtinimas)

Patvirtinimo primityvo tipas yra naudojamas sluoksnio teikiančio prašomą paslaugą, kad patvirtinti jog veikla yra užbaigta (sėkmingai arba nesėkmingai)

8.1.6.6 WDP protokolo aprašymas

8.1.6.6.1 Įžanga

Tam, kad įgyvendinti WDP datagramų protokolą, yra reikalinga:

- Tikslo prievadas (angl. *port*)

- Šaltinio prievadas

- Jei naudojamas nešėjas nepalaiko segmentavimo ir surinkimo, ši savybė yra įdiegiama WDP tiekėjo, nuo nešėjo priklausomu būdu.

8.1.6.6.2 WDP susiejimas su IP

Vartotojo datagramų protokolas UDP (angl. *User Datagram Protocol*) yra pritaikytas kaip WDP protokolo aprašymas bet kokiam belaidžiui tinklui, kur maršrutizavimo protokolas yra IP. UDP suteikia prievadais grįstą adresavimą, o IP suteikia segmentavimą ir surinkimą besujungiminėse datagramų paslaugose. Nėra prasmės nustatinėti naujus datagramų protokolus veikiančius naudojant IP kai jau įprastas vartotojo datagramų protokolas UDP suteikia tuos pačius mechanizmus ir funkcijas, ir jau yra plačiai realizuotas. Taigi visais atvejais kai IP protokolas yra galimas nešėjo paslaugose, WDP datagramų paslaugos siūlo nešėju naudoti UDP. UDP yra plačiai specifikuotas [17], kai tuo tarpu IP tinklo sluoksnis yra apibrėžtas [18][21].

Šie specifikacijoje aprašyti nešėjai naudoja UDP kaip WDP protokolo aprašymą:

GSM Circuit-Switched Data, GSM GPRS, ANSI-136 R-Data, ANSI-136 Circuit-Switched Data, GPRS-136, CDPD, CDMA Circuit-Switched Data, CDMA Packet Data, PDC Circuit-Switched Data,

PDC Packet Data, iDEN Circuit-Switched Data, iDEN Packet Data, PHS Circuit-Switched Data, TETRA Packet Data, DECT connection oriented / packet switched

8.1.6.6.3 WDP susiejimas su GSM SMS, ANSI-136 GHOST ir USSD

WDP nešėjai globalioje sistemoje mobiliam komunikavimui GSM (angl. *Global System for Mobile Communications*) yra GSM trumpųjų pranešimų paslauga (GSM SMS, angl. *Short Message Service*) ir GSM nestruktūrizuoti pridėtiniai paslaugų duomenys (GSM USSD, angl. *Unstructured Supplementary Service Data*). WDP nešėju ANSI-136 yra GSM Hosted SMS Teleservice (GHOST).

WDP per GSM ir ANSI-136 GHOST palaiko privalomą dvejetainį ir neprivalomą tekstinę antraštę. GSM USSD Phase 2 palaiko dvejetaines antraštes. SSM SMS Phase 2 ir ANSI-136 GHOST palaiko tiek privalomą dvejetainį ir neprivalomą tekstinę antraštes, o GSM SMS Phase 1 palaiko tekstines antraštes.

Kiekvienas paketas (segmentas) naudojamas WDP protokole yra identifikuojamas naudojant vartotojo duomenų antraštės informacijos elemento identifikatorių (angl. *User Data Header Information Element Identifier*), nurodantį sąsajos numerio struktūrą, esančią paketo antraštėje. Šis informacijos elemento identifikatorius GSM SMS, ANSI-136 GHOST ir USSD atlieka panašią funkciją, kaip protokolo identifikatorius (angl. *Protocol Identifier*) IP grįstuose tinkluose. Tai leidžia WDP protokolui egzistuoti kartu su kitomis savybėmis pasenusiuose nešančiuosiuose tinkluose.

8.1.6.6.3.1 Dvejetainis antraštės formatas

GSM SMS, ANSI-136 GHOST ir GSM USSD atveju WDP antraštės struktūra yra apibrėžta naudojant vartotojo duomenų antraštės UDH (angl. *User Data Header*) karkasą, kaip nurodyta specifikacijoje GSM0340:

2.3 lentelė WDP antraštės struktūra – UDH karkasas

Laukas	Ilgis
vartotojo duomenų antraštės ilgis	1 baitas
informacijos elemento identifikatorius „A“	1 baitas
informacijos elemento „A“ ilgis	1 baitas
informacijos elemento „A“ duomenys	Nuo 1 iki n baitų
informacijos elemento identifikatorius „B“	1 baitas
informacijos elemento „B“ ilgis	1 baitas
informacijos elemento „B“ duomenys	Nuo 1 iki n baitų
... ...	
informacijos elemento identifikatorius „n“	1 baitas
informacijos elemento „n“ ilgis	1 baitas
informacijos elemento „n“ duomenys	Nuo 1 iki n baitų

Lauke „informacijos elemento ilgis“ būna įrašomas skaičius rodantis kiek atitinkamas laukas „informacijos elemento duomenys“ užima baitų, pats laukas „informacijos elemento ilgis“ neįskaičiuojamas.

Lauke „vartotojo duomenų antraštės ilgis“ būna įrašomas skaičius rodantis kiek baitų užima „vartotojo duomenų antraštė“, pats laukas „vartotojo duomenų antraštės ilgis“ neįskaičiuojamas.

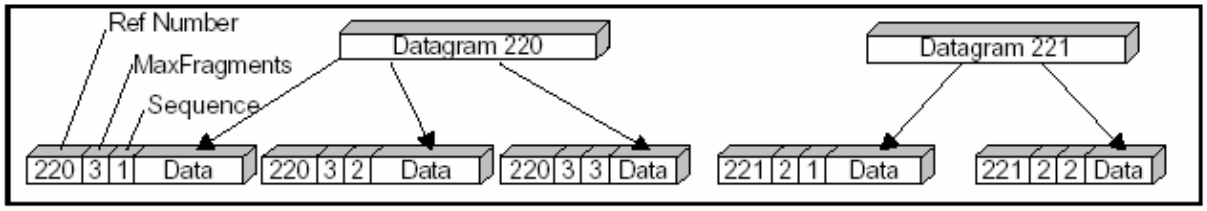
Skaičių tvarka yra „didžiausias baitas pirmas“. Tuo atveju jei naudojamas duomenų informacijos žodis kitoks nei baitas (oktetą), tada dvejetainė antraštė yra padidinama bitais prie informacijos žodžio pradžios pozicijos (GSM naudoja 7 bitų alfabetą) daugumoje atvejų. Tokiu būdu antraštė yra suderinama su senais įrenginiais, nepalaikančiais WDP datagramų protokolo.

8.1.6.6.3.2 Segmentacija ir surinkimas

WDP segmentacija yra įgyvendinta taip kaip ir specifikuota standarto GSM0340.

Yra apibrėžti du segmentacijos formatai – ilgasis ir trumpasis. Skirtumas tarp šių dviejų formatų yra tiktai datagramų nuorodos numerio (angl. *Datagram Reference Number*) maksimalus dydis. Formatas kai nuorodos numeriui naudojami tik 8 bitai yra tinkamas į mobilumą orientuotam komunikavimui, bet didelio masto taikomosioms programoms, orientuotoms į stacionarius serverius nuorodos numeriai greitai išnaudojami, ir vėl pradedami numeruoti nuo pradžios. Didesnė galima nuorodos numerių sritis stipriai sumažina riziką kad nuorodos numeriai persidengs, ko pasekmė būtų neteisingas surinkimas.

Mobiliosios stotys turi naudoti 8 ar 16 bitų nuorodų numerių antraštes pranešimų siuntimui, tačiau fiksuoti įrenginiai privalo naudoti 16 bitų nuorodos numerius, nebent aptinkama kad priimantis įrenginys palaiko tik 8 bitų nuorodos numerius. (šis aptikimas yra įgyvendinimo klausimas kiekvienam fiksuotų įrenginių gamintojui). Kiekvienas WDP įgyvendinimas privalo palaikyti tiek 8 tiek ir 16 bitų nuorodų numerių priėmimą, bet mobilusis įgyvendinimas gali būti apribotų galimybių – jam leidžiamas siųsti ir tik 8 bitų nuorodų numerius.



Datagramos segmentavimo schema

Ši schema parodo tipinį datagramos segmentavimą norint ją transportuoti. Ji rodo tik segmentacijos logiką, t.y. adaptacijos sluoksnį. Nuorodos numeris naudotas norint atskirti skirtingas datagramas. Segmentacijos ir surinkimo numeris naudoja eilės (angl. *sequence*) numerį ir fragmentų skaičiaus (angl. *Max Fragments*) numerį, tam kad apibrėžti pranešimo surinkimo eilę ir pilnumą.

Paketo antraštėje yra sekanti segmentacijos informacija:

1. WDP paketo nuorodos numeris (0-255, arba 0-65535)
2. maksimalus segmentų skaičius datagramoje (max. 255)
3. segmento numerį. (1-255)

Maksimalus segmentuotos datagramos dydis naudojant šią schemą priklauso nuo paketo dydžio. GSM SMS atveju didžiausias tinklo paketo dydis yra 140 baitų, o GSM USSD maksimalus tinklo paketo dydis yra 160 baitų. Eilės (nuorodos numerio ir segmento) numeris gali būti naudojamas norint nustatyti problemas su dublikuotais, pamestais, ir neteisinga tvarka atėjusiais paketais. Eilės numeris gali būti suprantamas kaip skaitiklis, kuris yra padidinamas su kiekvienu paketu. Surinkimas yra atliekamas naudojant gautų paketų sąrašą. Kai paketai gaunami, jie ta tvarka yra įdedami į sąrašą, o tada sąraše ieškoma pilnos datagramos (visi paketai atėjo, sutampantys eilės numeriai ir siuntėjo adresas). Jei tokia datagrama egzistuoja, ji gali būti perduota aukštesniajam sluoksniui.

8.1.6.6.3.2.1 Fragmentavimo informacijos elementas (trumpasis)

Trumpasis fragmentavimo informacijos elementas (angl. *Fragmentation Information-Element*) – identifikatorius kuris yra apibrėžtas standarto GSM0340, ten jis vadinamas sujungtų trumpųjų žinučių (angl. *Concatenated short messages*) 8 bitų eilės numeriu. Trumpasis informacijos elementas – identifikatorius yra baitas su šešioliktaine reikšme 00.

8.1.6.6.3.2.2 Fragmentavimo informacijos elementas (ilgasis)

Ilgasis fragmentavimo informacijos elementas (angl. *Fragmentation Information-Element*) – identifikatorius kuris yra apibrežtas standarto GSM0340, ten jis vadinamas sujungtų trumpųjų žinučių (angl. *Concatenated short messages*) 16 bitų eilės numeriu. Ilgasis informacijos elementas – identifikatorius yra baitas su šešioliktaine reikšme 08.

8.1.6.6.4 WDP susiejimas su CDMA SMS

WDP siunčia datagramas IS-637 SMS taškas-taškas žinučių vartotojo duomenų (angl. *User Data*) poparametryje. Datagramą sudaro keturių baitų antraštė, kurią seka duomenys. Kadangi kai kurios datagramos yra pernelyg ilgos kad tilptų SMS pranešime, datagramos gali būti segmentuojamos ir siunčiamos per keletą SMS žinučių, o tada surenkamos tikslo įrenginio. Standartas TIAEIA-637 neapibėžia segmentavimo ir surinkimų procedūrų, todėl jas apibrėšime čia. SMS pranešimai nešantys WDP datagramas turi naudoti WAP telepaslaugas, kurios yra aprašytos standarte TIAEIA-637.

8.1.6.6.4.1 Datagramų struktūra

WDP datagrama, turinti N baitų duomenų, siunčiama per IS-637 SMS turi sekančią struktūrą:

2.4 lentelė WDP datagramos siuntimas per SMS

Laukas	Ilgis (bitai)
Šaltinio prievadas (angl. *port*)	16
Tikslo prievadas (angl. *port*)	16
Duomenys	N * 8

8.1.6.6.4.2 SMS vartotojo duomenys

WDP SMS žinutes vartotojo duomenų poparametrio CHARi laukuose būna vienas WDP datagramos segmentas. CHARi laukų struktūra yra sekanti:

2.5 lentelė CHARi laukų struktūra

Laukas	Ilgis (bitai)
MSG_TYPE	8
TOTAL_SEGMENTS	8
SEGMENT_NUMBER	8
DATAGRAM	(NUM_FIELDS – 3) * 8

MSG_TYPE – Pranešimo tipas

Šis laukas turėtų būti nustatomas į „00000000“, kad parodyti jog tai yra WDP pranešimas. Šis laukas leidžia atskirti WDP pranešimus nuo kitų WAP pranešimų, tokių kaip WCMP pranešimai.

TOTAL_SEGMENTS – Viso segmentų

Esybė atliekanti segmentavimą IS-637 SMS nešėjui turi nustatyti šio lauko reikšmę į skaičių rodantį kiek viso segmentų sudarys pristatomą datagramą. Šis laukas nebus nustatomas į „00000000“.

SEGMENT_NUMBER – Segmento numeris

Esybė atliekanti segmentavimą IS-637 SMS nešėjui turi nustatyti šio lauko reikšmę į skaičių kelintas šis segmentas yra datagramoje. Pirmajam datagramos segmentui šis laukas bus nustatomas į „00000000“. Kiekviename sekančiame segmente laukas SEGMENT_NUMBER turi būti didinamas 1.

DATAGRAM – Datagramos baitai

Esybė atliekanti segmentavimą IS-637 SMS nešėjui turi užpildyti šį lauką atitinkamais datagramos segmento duomenų baitais. Vartotojo duomenų poparametrio NUM_FIELDS laukas turi būti nustatomas į datagramos baitų segmente skaičių, padidintą 3. Jei segmento numeris yra ne „00000000“, tai datagramos segmento baitų skaičius nebus didesnis nei prieš tai buvusio segmento baitu skaičius.

8.1.6.6.4.3 MESSAGE_ID lauko panaudojimas

Kai IS-637 SMS pranešime siunčiama WDP datagrama, IS-637 SMS galutinis taškas turi nustatyti pranešimo identifikatoriaus poparametrį MESSAGE_ID sekančiu būdu:

- Jei šiame SMS pranešime yra pirmoji WDP datagrama po to kai IS-637 SMS galinis taškas buvo perleistas, galinis taškas turi nustatyti MESSAGE_ID į atsitiktinį, nuo 0 iki 65535.

- Priešingu atveju, jei šis SMS pranešime yra pirmas WDP datagramos segmentas – galinis taškas turi padidinti MESSAGE_ID reikšmę vienetu lyginant su paskutine siųsta WDP datagrama moduliu 65536, taip sugeneruojant MESSAGE_ID lauką SMS pranešimui.

- Pranešimų centras nutraukiantis IS-637 SMS protokolą gali naudoti MESSAGE_ID kaip pranešimo eilės numerį, suteiktą trumpojo pranešimo esybes sąsajos protokolo. Tinkamo pranešimo eilės numerio pavyzdys yra „sar_msg_ref_num“ aprašytas [7]. Tačiau, jei eilės numeris nėra pateikiamas, pranešimų centras turi nustatyti MESSAGE_ID reikšmę pagal šiame skyriuje aprašytas taisykles.

- Jei SMS pranešime yra ne pirmas WDP datagramos segmentas, galinis taškas turi nustatyti MESSAGE_ID lauką į reikšmę, lygią pirmojo WDP datagramos segmento MESSAGE_ID lauko reikšmę.

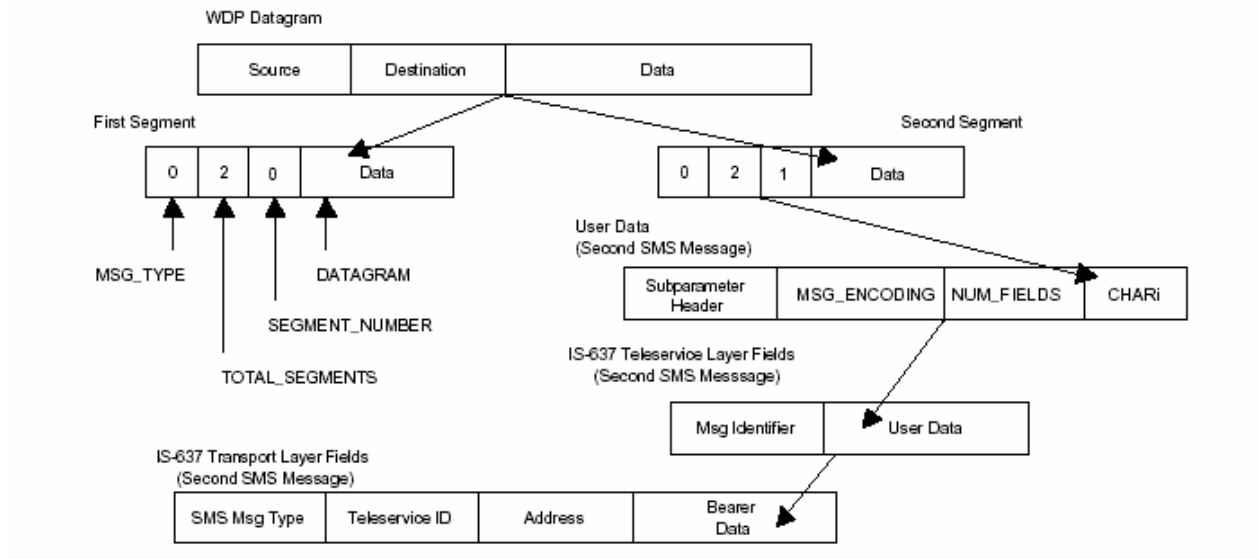
8.1.6.6.4.4 Segmentacija ir surinkimas

Datagramų segmentacija ir surinkimas naudoja 5 parametrus iš WAP SMS pranešimo: kilmės adreso parametrą iš SMS transporto sluoksnio, MESSAGE_ID iš SMS pranešimų identifikavimo poparametrių, MSG_TYPE, TOTAL_SEGMENTS, ir SEGMENT_NUMBER.

MSG_TYPE identifikuoja WAP pranešimus nešančius WDP datagramas. Kilmės adresas kartu su MESSAGE_ID identifikuoja datagramą. TOTAL_SEGMENTS ir SEGMENT_NUMBER yra naudojami tam kad patikrinti ar gauta pilna datagrama, t.y. ar ją jau galima perduoti aukštesniam lygiui.

8.1.6.6.4.5 Segmentacijos pavyzdys

Sekanti schema yra WDP datagramos, perduodamos per du IS-637 SMS taškas-į-tašką (angl. *Point-to-Point*) pranešimus, pavyzdys. Datagrama buvo padalinta į du segmentus, kurie yra pranešimų vartotojo duomenų poparametrių CHARi laukuose (čia matomas tik antras SMS pranešimas).



8.1.7 Belaidžio transporto sluoksnio saugumas WTLS

8.1.7.1 Įžanga

Sparčiai auganti belaidžio ryšio rinka didina mobiliųjų įrenginių, su papildoma verte / funkcijomis, poreikį. WAP (angl. *Wireless Application Protocol*) yra sukurtas norint patenkinti šį poreikį. WAP aprašo protokolų aibę pažangių mobiliųjų paslaugų kūrimui. Kadangi ši technologija yra vystoma tarptautinės organizacijos, vadinamo WAP Forum, tai laidžia mobiliojo ryšio vartotojui naudotis WAP teikiamomis paslaugomis visame pasaulyje, nepriklausomai nuo vietinės mobiliojo ryšio technologijos. WAP yra nepriklausoma nuo nešančiųjų paslaugų.

Belaidžio transporto sluoksnio saugumo protokolas WTLS (angl. *Wireless Transport Layer Security*) yra WAP architektūros saugumo sluoksnis. Pirminiai jo tikslai yra privatumo, duomenų integralumo ir autentiškumo nustatymo teikimas WAP taikymams. Saugumas yra reikalingas tam kad saugiau prisijungti prie tokių paslaugų kaip elektroninė komercija arba bankininkystė. Klientui ir serveriui turi būti atliktas autentiškumo nustatymas, ir tada susijungimas užkoduojamas. Perėmėjas-tarpininkas (angl. *Man-in-the-middle*) tipo atakoms kelias turi būti užkirstas, tam kad duomenys perdavimo metu nebūtų pakeičiami. Vartotojas nori būti tikras, kad naudojamos paslaugos yra būtent tos kurių jis tikisi. Tam tikrais atvejais paslaugos taip pat nori naudoti stiprius autentiškumo nustatymo mechanizmus su sertifikatais. Nors eteriu keliaujantys duomenys kai kuriuose tinkluose yra užkoduojamai, pilnas baigties-baigties (angl. *end-to-end*) saugumas nėra suteikiamas iš mobiliojo tinklo pusės. Tai ir yra pagrindinė priežastis kodėl WTLS yra reikalingas.

WTLS teikia transporto paslaugų sąsajas aukštesniam sluoksniui. Ši sąsaja yra panaši į transporto paslaugų sąsajas esančias „žemiau“ WTLS. WTLS yra pagrįstas gerai žinomu TLS v1.0 saugumo sluoksniu, plačiai paplitusiu internete. Žinoma dėl belaidžių tinklų prigimties buvo reikalingos modifikacijos ir pakeitimai. Belaidžiai tinklai reikalauja tiek datagramomis tiek ir susijungimais orientuoto transporto sluoksnio protokolų. Mobilioji įranga taip pat iškelia reikalavimus algoritmams, nes apdorojimo galia ir turimos atmintinės kiekis yra ribotas. Be to reikia įvertinti mažą pralaidumą, bei apribojimus kriptografijos panaudojime ir eksportavime.

WTLS savyje naujų savybių, tokių kaip datagramų palaikymas, optimizuotas paketo dydis ir pasisveikinimas (angl. *handshake*), dinaminis raktų atnaujinimas. Jis buvo optimizuotas mažo pralaidumo nešantiesiems tinklams, turintiems palyginus didelį vėlinimą. Mobilioji įranga, tokia kaip mobilieji įrenginiai, gali būti sukonstruota palaikyti tik tam tikrą šifravimo rinkinių aibę,

Šio skyriaus tikslas yra įvertinti saugumą, naudojamą WTLS versijoje 1.1. Svarbiausias klausimas – „Ar teikiamas saugumo lygis yra pakankamas?“. Nors belaidžiai tinklai ir sukuria daug

reikalavimų, teikti priimtiną saugumo lygį įmanoma. Pavyzdžiui saugumo protokolas kuris leidžia įsibrovėliui slapta matyti perduodamus duomenis yra nepriimtinas. Tačiau kita vertus, absoliutaus saugumo ir gero patogumo derinys net teoriškai nėra įmanomas. Saugumo problemos kurios jau yra aptiktos WTLS yra modifikacijų pasekmė. Laimei WTLS vis dar plėtojamas, ir visos problemos ištaisomos, suteikiant vis daugiau saugumo. Žinoma visapusis saugumas priklauso ir nuo kitų faktorių, ne tik nuo WTLS.

8.1.7.2 Duomenų perdavimo saugumas

Duomenų perdavimo saugumas susideda iš mažesnių saugumo esybių. Sekančiuose skyreliuose bus pristatomos ir analizuojamos skirtingos saugumo esybės, kaip WTLS analizės kriterijai.

8.1.7.2.1 Privatumas

WAP (angl. *Wireless Application Protocol*) specifikacija apibrėžia privatumą, taip kad naudojamas duomenų perdavimo metodas užtikrina privatų baigtis-baigtis (angl. *end-to-end*) duomenų perdavimą. Jis turi būti nesuprantamas tokiems tarpiniams asmenims galbūt perėmusiems duomenų srautą.

Privatumo išlaikymas iš esmės yra kova prieš atskleidimo grėsmę. Pagrindinis įrankis norint sėkmingai atlikti šią užduotį – kriptografija. Paprastas tekstas tiesiog užkoduotas, o tada atkoduotas tam kad įgyvendint privatumą. Jei paprastas tekstas yra užkoduojamas gerą kodavimą, pasidaro beveik neįmanoma įsibrovėliui atkoduoti ir perskaityti originalų turinį. Pagrindinis reikalavimas saugiam užkodavimui yra bendros slaptos informacijos, o ne algoritmo, naudojimas. Erdvė, iš kurios bendra slapta informacija yra pasirenkama, turi būti didelė. Dar daugiau – naudojamas kriptografinis metodas turi sukurti rezultatą kuris atrodo atsitiktinis visiems statistiniams testams. Ir galiausiai – metodas turi būti atsparus visiem žinomiems atakos tipams.

Tačiau užkoduoti duomenys yra beverčiai, jei gavėjas negalės jų atkoduoti. Siuntėjas ir gavėjas turi turėti bendrą metodą duomenų užkodavimui ir atkodavimui. Abi pusės turi žinoti naudojamą kriptografijos būdą, bei bendrą slaptą informaciją. Bendra slapta informacija yra informacija kurią žino abi pusės ir niekas daugiau.

Tačiau yra ir kitokio tipo privatumas. Ne visada būna taip kad informacija turi gavėją. Kartais informacija niekada nebus kieno nors atkoduojama, pavyzdžiui slaptažodžiai. Šis būdas yra vadinamas vienos krypties užkodavimas. Kitaip sakant – nėra būdo užkoduotiems duomenims atkoduoti, ir gauti pradinę informaciją. Galima tik užkoduoti turimą informaciją, ir palyginti ar ją užkodavus, rezultatas sutampa su anksčiau užkoduotos informacijos. Hash reikšmės yra vienas dažniausiai pasitaikančių vienos krypties užkodavimo metodų.

8.1.7.2.2 Autentiškumo nustatymas

Dažnai naudojama B. Schneider autentiškumo nustatymo sąvoka sako, kad gavėjas turi garantuotai žinoti pranešimo siuntėją, o trečioji šalis turi negalėti apsimesti šiuo siuntėju.

Autentiškumo nustatymas yra metodika, teigiamas tapatumas teisingumui nustatyti. Pradžioje viena pusė „prisistato“ ir nurodo savo tapatumą. To neužtenka. Pusė prie kurios jungiamasi taip pat turi garantuotai žinoti kad besijungianti pusė yra būtent ta, kuria prisistato. Todėl pusė prie kurios jungiamasi turi turėti būdą kaip patikrinti tapatumą. Tai gali būti toks paprastas būdas kaip slaptažodis, arba šiek tiek sudėtingesni – skaitmeninis parašas ar sertifikatas. Bet besijungianti pusė taip pat nori būti tikra kad ta pusė prie kurios yra jungiamasi yra teisinga. Todėl pusė prie kurios jungiamasi taip pat turi pateikti tam tikrą savo tapatybės įrodymą.

Po autentiškumo nustatymo, paslaugų tiekėjas gali būti tikras kad paslauga yra teikiama vartotojui turinčiam teisies šia paslauga naudotis. Iš kitos pusės – vartotojas gali būti tikras dėl paslaugų tiekėjo.

8.1.7.2.3 Originalumas

Dažnai naudojama B. Schneider originalumo (angl. *integrity*) sąvoka sako, kad pranešimo gavėjas turi galėti patikrinti ar pranešimas nebuvo modifikuotas, o įsibrovėlis turi negalėti pateikti pakeistą pranešimą vietoje originalaus.

Originalumo išlaikymas reiškia informacijos patikimumo saugumą. Reikia būdo kaip išvengti neautorizuotų pakeitimų, arba bent jau aptikti šios pakeitimus. Originalumas yra užtikrinamas skaičiuojant kontrolines sumas originaliai informacijai. Žinoma tiesiog kontrolinių sumų skaičiavimo neužtenka. Į skaičiavimus taip pat turi būti įtraukta informacija susijusiu su siuntėju, pvz. yra naudojamas vartotojo skaitmeninis parašas.

Daugumoje atvejų, originalumo išsaugojimas yra svarbiau nei privatumo išsaugojimas. Paprastai daug svarbiau yra tai, kad gaunama informacija būtų nepakeista, nei kad informacija būtų pakeista kitų šalių, net jei jos informacijos pilnai ir neperėmė / neatkodavo. Pavyzdžiu galima paimti banko pavedimus. Negerai jei kas nors sužinos kiek tam tikras vartotojas turi pinigų, tačiau bus daug blogiau, jei šie pinigai bus pavogti.

8.1.7.2.4 WTLS saugumas?

WTLS yra skirtas teikti saugumą WAE (angl. *Wireless Application Environment*). Belaidžiai tinklai iškelia naujus iššūkius saugumo architektūros įdiegimui, lyginant su tradiciniais, susijungimais pagrįstais modeliais, kurie yra naudojami internete.

Sekančio skyriaus tikslas ir yra nustatyti ar WTLS atitinka saugumo reikalavimus pateiktus ankstesniuose skyriuose.

8.1.7.3 Belaidžio transporto sluoksnio saugumas

Saugumas WAP architektūroje turėtų leisti paslaugas teikti per potencialiai nesaugius mobiliuosius tinklus, tuo pačiu išlaikant duomenų originalumą. Paslaugos atmetimo DOS (angl. *denial of service*) taip pat reikia išvengti. Belaidžiai mobilieji tinklai įneša daug reikalavimų saugumo sluoksniui, todėl egzistuojantys saugumo protokolai negali būti panaudojami mobiliuosiuose tinkluose jų protokolų) specialiai nepritaikius.

Vienas iš svarbiausių reikalavimų yra žemo duomenų perdavimo greičio palaikymas. Pavyzdžiui SMS kaip nešėjas gali būti tikrai labai lėtas – tik 100 bitų per sekundę. Todėl papildomai perduodamų duomenų dydis turi būti kaip įmanoma mažas. Lyginant su įprastu standartu jau tapusiu TLS (angl. *Transport Layer Security*), anksčiau dar žinomu SSL (angl. *Secure Socket Layer*) vardu, datagramų transporto sluoksnis dėl belaidžių mobiliųjų tinklų prigimties taip pat turi būti palaikomas. Protokolas turi suvaldyti dingusias, susidvejinusias, ar neteisinga tvarka atėjusias datagramas, tuo pačiu nenutraukdamas susijungimo būsenos.

Kitos problemos su kuriomis tenka susidurti – lėta sąveika, maža apdorojimo galia bei atmintinės talpa. Taip pat yra ir kitų apribojimų susijusių su eksportavimu bei kriptografija. Perdavimo pirmyn-atgal (angl. *round-trip*) laikas gali būti ilgas, ir dėl to neturėtų būti uždaromi susijungimai. Pavyzdžiui laikas tarp užklausos ir atsakymo naudojant SMS nešėją gali užtrukti net 10 sekundžių. Naudojami kriptografijos algoritmai turi būti pakankamai nesudėtingi, tam kad mobilieji terminalai galėtų juos vykdyti. Palaikomų kriptografinių algoritmų skaičius turi būti kiek įmanoma sumažintas, o palikti reikia tuos kurie yra nedidelės atminties. Taip pat reikia atsižvelgti į mobiliųjų įrenginių turimą operatyviosios atmintinės (RAM) kiekį. Taip pat reikia numatyti tai, kad kai kuriose šalyse negalima naudoti tam tikrų gero užkodavimo algoritmų. Todėl tokiose šalyse visą laiką turi būti naudojamas maksimalus leistinas saugumo lygis. Taip pat yra skirtumų tarp leidžiamo užkodavimo ir autentiškumo nustatymo. Daugeliu atveju geras autentiškumo nustatymas yra leidžiamas, tačiau kodavimas – ne.

Taigi WTLS tikslas yra lengvasvoris efektyvus protokolas, kuris kurtas atsižvelgiant ribotą pralaidumą, atmintinę, bei apdorojimo galią.

8.1.7.3.1 Specifikacija

WTLS sluoksnis veikia virš transporto protokolo sluoksnio, ir teikia aukštesniam WAP sluoksniui saugaus transporto sąsają. Sąsaja išlaiko žemiau esančią transporto sąsają bei suteikia papildomus metodus valdyti saugiems susijungimams.

Taigi WAP, naudodamas WTLS tiekia baigties-baigties (angl. *end-to-end*) saugumą tarp WAP protokolo baigties taškų (angl. *endpoints*). Baigties taškai iš tikro yra mobilusis terminalas ir WAP vartai (angl. *gateway*). Kai WAP vartai kilmės serveriui teikia užklausą, bus naudojamas SSL po

HTTP, tam kad apsaugoti užklausą. Tai reiškia jog duomenys yra atkoduojami ir vėl užkoduojami WAP vartuose.

Pilnai saugus susijungimas tarp kliento ir serverio gali būti pasiektas dviem skirtingais būdais. Saugiausias būdas yra paslaugos tiekėjui pastatyti WAP vartus atskirame tinkle. Tada visas susijungimas tarp kliento ir paslaugos gali būti laikomas saugiu, nes atkodavimas vyks tiktai perdavimui pasiekus tiekėjo tinklą, o ne mobiliojo operatoriaus tinkle. Kitas variantas yra WAP vartų funkcionalumą įdiegti į kilmės (tikslo) serverį. Tai yra saugiausias galimas būdas.

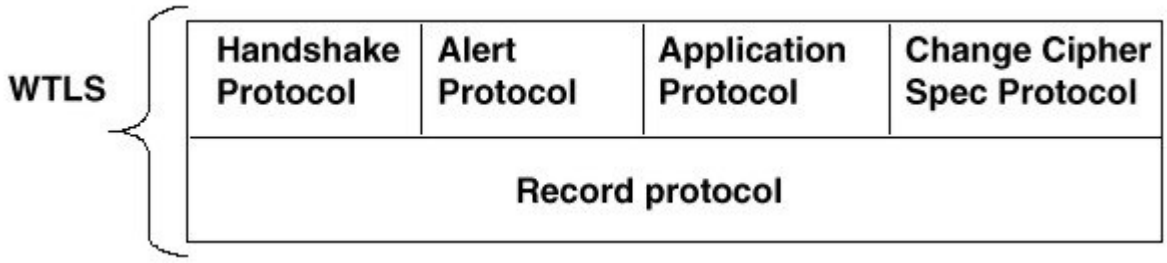
Paslaugų bei turinio tiekėjai gali pasitikėti mobiliųjų operatorių vartais, ir naudoti virtualius privačius tinklus VPN kad sujungti jų serverius su WAP vartais. Tačiau tada nebus galimybės valdyti ir kontroliuoti WTLS naudojamų parametrų WAP vartuose.

Besijungiančios pusės gali nuspręsti kokias saugumo savybes norės naudoti susijungimo metu. Pagal saugumo reikalavimus programinė įranga gali įjungti arba išjungti WTLS galimybes. Pavyzdžiui privatumas gali būti išjungtas, jei tinklas jau siūlo šią paslauga žemesniame lygyje. Susijungimas tarp dviejų terminalų taip pat gali būti saugomas WTLS.

8.1.7.3.2 WTLS vidinė architektūra

WTLS įrašų (angl. *record*) protokolas yra sluoksniuotas protokolas kuris priima neapdirbtus duomenis iš aukštesnių sluoksnių siuntimui, ir jiems pritaiko suspaudimo bei užkodavimo algoritmus. Dar daugiau – įrašų protokolas pasirūpina duomenų originalumu ir autentiškumo nustatymu. Priimti duomenys būna atkoduojami ir atspaudžiami, o tada perduodami aukštesniems sluoksniams.

Įrašų protokolas yra dalinamas į keturis protokolus – klientus. Protokolo dėklas yra parodytas schemoje žemiau. Skirtingi klientai yra aprašyti sekančiuose gilesniuose skyriuose. Taikymų protokolas (angl. *application protocol*) čia neaptašomas, nes jis teikia sąsają aukštesniems sluoksniams.



WTLS vidinė architektūra

8.1.7.3.2.1 Kodavimo pakeitimo protokolas

Kodavimo pakeitimo (angl. *Change Cipher Spec*) yra siunčiamas sent kitai pusei arba kliento arba serverio. Kai gaunamas kodavimo pakeitimo pranešimas, pranešimo siuntėjas nustato dabartinę laukimo būseną į „laukiamas“, o gavėjas nustato dabartinę nuskaitymo būseną į „laukiamas“.

Kodavimo pakeitimo pranešimas yra siunčiamas pasisveikinimo (angl. *handshake*) fazėje po to, kai susitarimas kokie bus naudojami saugumo parametrai, yra priimtas.

8.1.7.3.2.2 Perspėjimo protokolas

Įrašų protokolas taip pat tiekia perspėjimo pranešimų turinio tipą. Yra trijų tipų perspėjimo pranešimai: įspėjimai, kritiniai ir lemtingi (angl. *fatal*). Perspėjimo pranešimai yra siunčiami naudojant dabartinę saugumo būseną, pvz. suspaustas ir užkoduotas, arba be kompresijos ar kodavimo.

Jei perspėjimo pranešimas priklausantis lemtingų tipui yra išsiųstas, abi pusės nutraukia saugų susijungimą. Kiti susijungimai naudojantys saugią sesiją gali tęsti, tačiau sesijos identifikatorius turi būti padarytas negaliojančiu, kad nutrūkęs susijungimas nebūtų panaudotas naujo saugus susijungimo sukūrimui.

Kritiniai dabartinės būsenos pranešimai taip pat reiškia tai, kad abi pusės nutraukia saugų susijungimą. Kiti susijungimai naudojantys saugią sesiją gali tęsti, o sesijos identifikatorius neprivalo būti padarytas negaliojančiu, taigi nutrūkęs susijungimas gali būti panaudotas naujo saugus susijungimo sukūrimui.

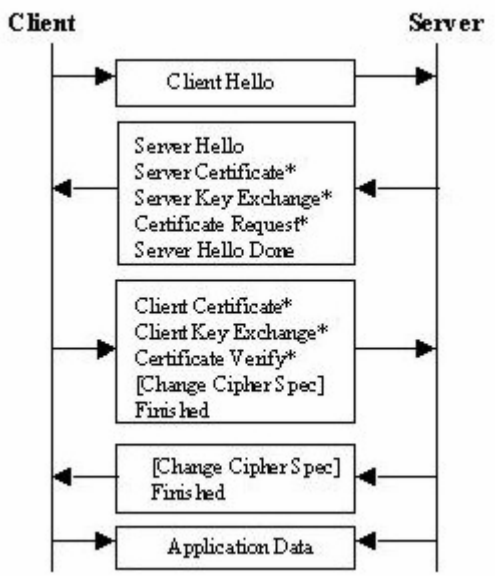
Taigi susijungimas yra uždaromas naudojant perspėjimo pranešimus. Bet kuri pusė gali inicijuoti apsikeitimą užbaigimo žinutėmis. Jei gaunama užbaigimo žinutė, visi duomenys gauti po jos bus ignoruojami. Taip pat reikalaujama kad pranešimą apie uždarymą gavusi pusė atsiliepdama ne tik patvirtintų, bet tuo pačiu ir patikrintų ar jis tikrai buvo siųstas.

Klaidų valdymas WTLS yra pagrįstas perspėjimo pranešimais. Kai aptinkama klaida, aptikusi pusė siunčia pranešimą su atitinkama klaida. Tolesnės procedūros priklauso nuo įvykusios klaidos.

8.1.7.3.2.3 Pasisveikinimo protokolas

Visi su saugumu susiję parametrai yra suderinami pasisveikinimo (angl. *handshake*) metu. Tarp šių parametrų yra naudojamos protokolų versijos, naudojami kriptografiniai algoritmai, informacija apie naudojama autentiškumo nustatymą ir viešojo rakto generavimo metodikas (skirtas generuoti „bendrai slaptai informacijai“).

Sekančioje schemoje matoma pasisveikinimo veikimo schema.



Pilno pasisveikinimo schema

Pasisveikinimas (angl. *handshake*) prasideda pasveikinimo (angl. *hello*) pranešimu. Klientas siunčia pasveikinimo pranešimą serveriui. Serveris turi atsakyti serverio pasveikinimo pranešimu. Šiais dviem pranešimais komunikuojančios pusės susitaria kokios bus sesijos galimybės. Pavyzdžiui klientas praneša palaikomus užkodavimo algoritmus, bei naudojamus sertifikatus. Serveris atsako, pasirinkdamas ir nustatydamas parametrus visai sesijai, iš tų kuriuos siuntė klientas. Jei klientas nepasiūlė norimų savybių – serveris turi nuspręsti kurias naudoti.

Po to kai klientas išsiuntė pasveikinimo pranešimą, jis priiminėja pranešimus iš serverio kol gauna pranešimą kad pasveikinimas baigtas. Serveris taip pat išsiunčia serverio sertifikato pranešimą, jei autentiškumo nustatymas yra reikalingas iš serverio pusės. Dar daugiau – serveris gali reikalauti kad autentiškumą patirtintų klientas. Serverio rakto apsikeitimas (angl. *Server Key Exchange*) yra naudojamas norint klientui suteikti viešąjį raktą, kuris galės būti naudojamas darbui, arba „pradinės bendros slaptos informacijos“ perdavimui.

Po to kai iš serverio priimamas pasveikinimo baigimo pranešimas, klientas tęsia pasisveikinimą. Jei paprašytas, klientas siunčia kliento sertifikato pranešimą, kuriuo nurodo savo autentiškumą. Tada klientas išsiunčia kliento rakto apsikeitimo (angl. *Client Key Exchange*) pranešimą, kuriame yra „pradinės bendra slapta informacija“ užkoduota naudojant serverio viešąjį raktą arba informacija kad abi pusės gali baigti apsikeitimą raktais. Galiausiai, klientas siunčia užbaigimo pranešimą, kurio patvirtina visų prieš tai siųstų duomenų, įskaitant skaičiuojamo saugumo informaciją.

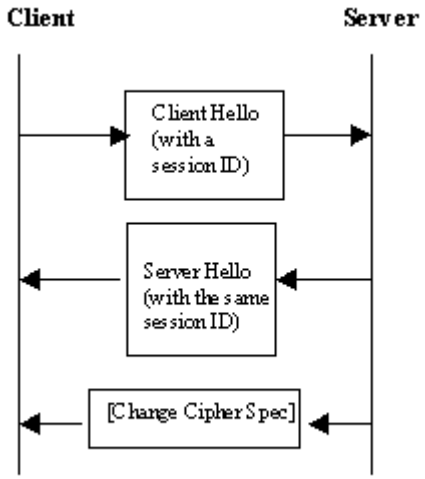
Serveris turi atsakyti užbaigimo pranešimu, kur jis taip pat patvirtina apsikeitimo bei skaičiuojamo saugumo informaciją. Papildomai, abi pusės turi išsiųsti kodavimo pakeitimo pranešimą. To pasekoje pusės žino kad reikia pradėti naudoti susitartus sesijos parametrus.

Jei klientas ir serveris nusprendžia pratęsti prieš tai suderintą sesiją, pasisveikinimas gali būti pradėtas siunčiant kliento pasveikinimo pranešimą, kur sesijos identifikatorius yra nustatomas į

naudotą ankstesnėje sesijoje. Jei abi pusės turi bendrą sesijos identifikatorių, jos gali tęsti saugios sesijos naudojimą.

Abi pusės baudoti susijungimą gali po to kai jos patvirtina sesiją, ir informuoja viena kitą išsiųsdamos kodavimo pakeitimo pranešimą.

Žemiau pateikiama schema kai klientas ir serveris nusprendžia pratęsti prieš tai suderintą sesiją.

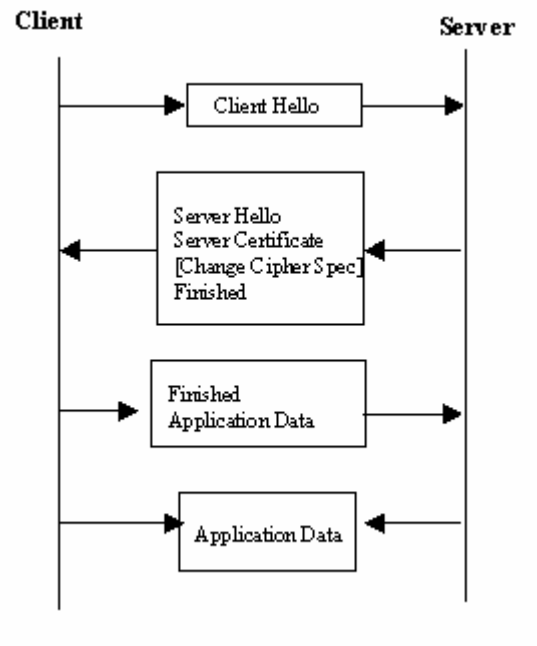


Pasisveikinimo schema naudojant susijungimo pratęsimą

WTLS taip pat apibrėžia sutrumpintą pasisveikinimą, kur siunčiami tik pasveikinimo ir užbaigimo pranešimai. Šiuo atveju abi pusės privalo turėti „bendrą slaptą informaciją“, kuri naudojama kaip pradinė.

Kitokia, optimizuota, pilno pasisveikinimo variacija yra kai serveris gali priimti kliento sertifikatą naudodamas patikimą trečiąją šalį, remiantis informacija pateikta kliento išsiųstame pasveikinimo pranešime. Naudojant informacija kuri pateikta abiejų pusių sertifikatuose, abi pusės gali užpildyti „bendros slaptos informacijos“ reikšmes naudojant Diffie-Hellman raktų apsikeitimo metodiką.

Serveris tusi išsiųsti serverio pasveikinimą, sertifikatą ir užbaigimo pranešimą klientui, tam kad užbaigti pasisveikinimą iš serverio pusės. Klientas tuo tarpu atsako kliento užbaigimo pranešimu.



Optimizuota pilno pasisveikinimo schema

8.1.7.3.3 Autentiškumo nustatymas

WTLS autentiškumo nustatymas yra atliekamas sertifikatų pagalba. Autentiškumo nustatymas gali įvykti tarp kliento ir serverio, arba tik klientas nustato serverio autentiškumą. Antroji procedūrą galima tik tuo atveju jei serveris tai leidžia. Serveris gali reikalauti kad klienas nurodytų savo autentiškumą serveriui. Tiesa WTLS specifikacija autentiškumo nustatymą apibrėžia kaip neprivalomą procedūrą.

Šiuo metu X.509v3, X9.68 ir WTLS sertifikatai yra palaikomi. Kai WTLS specifikacijos versija 1.1 buvo išleista, X9.68 sertifikatas dar nebuvo apibrėžtas. WTLS sertifikatas yra optimizuotas dydžiui.

Autentiškumo nustatymo procedūra seka iškart po kliento ir serverio pasveikinimo pranešimų. Kai autentiškumo nustatymas yra naudojamas, serveris išsiunčia serverio sertifikato pranešimą klientui. Sekančioje lentelėje yra sertifikuotos informacijos kurią pateikia serveris sąrašas.

2.6 lentelė Serverio sertifikato pateikiama informacija

Informacija	Paaiškinimas
Sertifikato versija	Sertifikato versija
Parašo algoritmas	Algoritmas naudotas to pasirašyti sertifikatą
Leidėjas	Nurodo kas pasirašė sertifikatą, paprastai sertifikatų agentūra
Negalioja iki	Sertifikato galiojimo pradžios data
Negalioja po	Sertifikato galiojimo pabaigos data
Objektas	Rakto savininkas, susijęs su sertifikuotu viešuoju raktu
Viešo rakto tipas	Viešo rakto tipas (algoritmas)
Parametrų aprašas	Aprašo su šiuo raktu susijusius parametrus

Informacija	Paaiškinimas
Viešasis raktas	Viešasis raktas, kuris yra sertifikuojamas

Faktiškai, priimančioji pusė gauna sertifikatų sąrašą. Sąrašas yra sertifikatų grandinė, kur pirmasis yra paties serverio sertifikatas. Kiekvienas sekantis sertifikatas sertifikuoja ankstesnįjį. Pagal WTLS specifikacija, norint optimizuoti duomenų srautą ir apdorojimą kliento pusėje, serveris gali siųsti ir tik vieną sertifikatą – serverio sertifikatą pasirašytą specifikacijos agentūros viešuoju raktu, kuris yra platinamas nepriklausomai.

Serveris gali taip pat pasiųsti sertifikato užklausos pranešimą klientui, tam kas nustatyti jo autentiškumą. Pranešimas iš karto seka serverio sertifikato pranešimą, ir serverio raktų apsikeitimo pranešimą (jei jis siunčiamas). Savaime suprantama, sertifikato užklausos pranešimas klientui siunčiamas tik tada, jei serveris išsiunčia ir serverio sertifikato pranešimą. Pranešime serveris išvardina visas priimtinas sertifikatų agentūras. Jei sąrašas tuščias – klientas gali siųsti bet kokį sertifikatą.

Užklaustas, klientas siunčia kliento sertifikato pranešimą atgal serveriui. Kliento pusės sertifikatai naudoja tą pačią struktūrą kaip ir serverio sertifikatai. Jei klientas neturi tinkamo sertifikato, klientas turi išsiųsti tuščią sertifikato pranešimą. Taip pat klientas gali išsiųsti lemtingos pasisveikinimo klaidos perspėjimo pranešimą, ir uždaryti saugų susijungimą. Kliento sertifikato pranešimas paprastai savyje turi keletą sertifikatų. Tai yra priimtina, nes sertifikatų sąrašą apdoros serveris, kuris paprastai turi gerokai daugiau apdorojimo galios nei klientas.

Tikslus patikrinimas yra atliekamas kliento, jei kliento sertifikato pranešimas yra išsiunčiamas. Klientas sujungia visus gautus iš serverio ar savo sukurtus pranešimus, ir apskaičiuoja hash reikšmę pasirašymui.Šis pranešimas yra išsiunčiamas serveriui, kuris gali įsitikinti jos autentiškumo nustatymas kol kas praėjo gerai.

8.1.7.3.4 Apsikeitimas raktais

Tam kad užtikrinti saugų komunikavimo kanalą reikia saugiai apsikeisti užkodavimo raktais, arba pradinėmis reikšmėmis raktų skaičiavimui. Sertifikuotas apsikeitimas viešaisiais raktais buvo aprašytas ankstesnėje dalyje. Tačiau yra įmanomas ir toks atvejis, kai serverio sertifikato pranešime nėra pakankamai duomenų kad klientas galėtų keistis „pradine bendra slapta informacija“ (pradinė bendra slapta informacija yra informacija reikalinga apskaičiuoti bendrai slaptai informacijai). Šiuo atveju serverio rakto apsikeitimo pranešimas yra naudojamas tokiai informacijai suteikti.

WTLS raktų apsikeitimo mechanizmas taip pat tiekia anonimišką būdą raktų apsikeitimui. Šios procedūros metu serveris siunčia serverio rakto apsikeitimo pranešimą, kuriame yra viešasis serverio raktas. Taktų apsikeitimo algoritmas gali būti RSA, Diffie-Hellman, arba elipsinių kreivių Diffie-Hellman. Pranešime nebūna jokios sertifikuotos informacijos.

Tiek RSA tiek anoniminio RSA atveju klientas užkoduoja „pradinę bendrą slaptą informaciją“ naudodamas serverio viešąjį raktą ir nusiunčia jį serveriui kliento rakto apsikeitimo pranešime. Su Diffie-Hellman grįstais algoritmais klientas ir serveris apskaičiuoja „pradinę bendrą slaptą informaciją“ pagrįstą savo privačiu raktu ir kitos pusės viešuoju. Šis pranešimas yra praleidžiamas jei koks nors Diffie-Hellman grįstas algoritmas buvo naudojamas, ir kliento algoritmas buvo užklaustas, bei klientas jį pateikė.

Jei klientas pateikia savo palaikomų kriptografinių raktų apsikeitimo mechanizmų sąrašą, serveris gali pasirinkti ar jis naudos kliento pasiūlytus, ar kitus. Jei klientas nepasiūlė metodų – serveris turi tai padaryti.

8.1.7.3.5 Privatumas

WTLS privatumas yra įgyvendintas remiantis komunikacijos kanalo užkodavimu. Naudojamais užkodavimo metodais ir visomis skaičiavimui reikalingomis reikšmės yra apsikeičiama pasisveikinimo metu.

Pirmuose pranešimuose, kliento bei serverio pasveikinimuose, apsikeičiama ir atsitiktinėmis reikšmėmis. Vėlesnėse fazėse klientas ir serveris apsikeičia „pradine bendra slapta informacija“. Ši reikšmė yra perduodama per saugų kanalą kaip aprašyta ankstesnėje dalyje. Šios reikšmės yra naudojamos kai reikia apskaičiuoti „bendrą slaptą informaciją“. „Bendra slapta informacija“ yra 20 baitų seka, kuri yra skaičiuojama naudojant sekančią formulę: BSI = PRF (PBSI, „BSI“, Kliento_Pasveikinimo.Atsitiktinė_Reikšme + Serverio_Pasveikinimo.Atsitiktinė_Reikšmė)[0..19];

Čia PRF – pseudo-atsitiktinė funkcija, kuri kaip parametrus paima slaptą informaciją, „pradžios tašką“ (angl. *seed*), identifikacinę žymą ir gražina norimo ilgio rezultatą.

Naudojamas užkodavimo algoritmas yra pasirenkamas serverio pasveikinimo pranešime. Šiame pranešime serveris informuoja klientą kad jis pasirinko vieną kodavimo rinkinį. Klientas serveriui pateikia kodavimo rinkinių sąrašą. Kodavimo rinkiniai susideda iš „*bulk*“ duomenų kodavimo algoritmų bei „MAC“ algoritmo. Pirmas sąrašo elementas rodo kliento pageidaujamą algoritmą. Jei serveris sąraše neranda jam tinkamo kodavimo algoritmo, pasisveikinimas nepavyksta ir susijungimas uždaromas.

Šiuo metu dažniausiai naudojami „*bulk*“ kodavimo algoritmai yra palaikomi, tokie kaip RC5 su 40, 56 ir 128 bitų raktais, DES su 40 ir 56 bitų raktais, 3DES ir IDEA su 40, 56 ir 128 bitų raktais. Visi algoritmai yra blokų kodavimo algoritmai, o srautų kodavimo algoritmai yra nepalaikomi.

Užkodavimo raktai yra tvarkomi naudojant rakto bloką. Rakto blokas yra paskaičiuojamas iš pradinių pasisveikinimo metu perduotų reikšmių.

```
pasisveikinimo_blokas =
PRF(
     BSI +
     identifikacinė_žyma +
```

```
        eilės_numeris +
        Kliento_Pasveikinimo.Atsitiktinė_Reikšme +
        Serverio_Pasveikinimo.Atsitiktinė_Reikšmė
    );
```

Rakto blokas yra priklausomas nuo eilės numerio, kuris daro rakto bloką kintantį. Rakto blokas yra perskaičiuojamas tam tikrais intervalais, priklausomais nuo rakto atnaujinimo dažnio. Rakto atnaujinimo dažnis yra nusprendžiamas kliento ir serverio pasisveikinimo pranešimų metu. Identifikacinė žyma yra tiesiog tekstinė reikšmė skaičiavimui. Klientas naudoja tekstą „*client expansion*“, o serveris – „*server expansion*“.

Užkodavimo raktas, pradinis vektorius ir MAC „slapta informacija“ yra gaunama iš rakto bloko, remiantis raktų ilgiais reikalingais pasirinktiems algoritmams.

8.1.7.3.6 Originalumas

Duomenų originalumas (angl. *integrity*) yra užtikrinamas naudojant pranešimų autentiškumo nustatymo kodus MAC (angl. *message authentication codes*). Naudojami MAC algoritmai yra apsprendžiami tuo pačiu metu kaip ir kodavimo algoritmai. Klientas siunčia palaikomų MAC algoritmų sąrašą, kur pageidaujamas algoritmas yra sąrašo pradžioje. Serveris gražina pasirinktą algoritmą serverio pasveikinimo pranešime.

WTLS palaiko dažnai naudojamus MAC algoritmus, tokius kaip SHA ir MD5. Yra keletas skirtingų abiejų algoritmų versijų, pavyzdžiui SHA egzistuoja su 0, 40 ir 80 bitų MAC dydžiais. MAC su raktais yra skaičiuojami naudojant SHA-1. Modifikuoti algoritmai yra pagrįsti SHA-1, bet tik dalis rezultato yra naudojama. Egzistuoja tokios pačios MD5 algoritmo modifikacijos.

Specialus MAC algoritmas yra SHA_XOR_40, kuris yra a 5-baitų kontrolinė suma. Iš pradžių informacija yra padalinama į 5-baitų blokus. Tada visiem blokams iš eilės yra atliekama operacija XOR. Reikalaujama kad XOR MAC būtų užkoduotas, ir yra naudojamas tik CBC metodo blokų kodavimuose. Algoritmas yra skirtas įrenginiams su ribotomis procesoriaus galimybėmis.

MAC yra generuojamas naudojant suspaustus WTLS duomenis. Sekančios reikšmės yra naudojamos skaičiuoti MAC:

```
HMAC_Hash = (MAC_SI, eilės_numeris + WTLS_Suspaustų_Duomenų.įrašo_tipas +
WTLS_Suspaustų_Duomenų.duomenų_ilgis + WTLS_Suspaustų_Duomenų.fragmentas);
```

HMAC_Hash demonstruoja naudotą MAC algoritmą su raktu, t.y. SHA-1 ar MD5. MAC_SI yra rakto bloko reikšmių. Po to kai HMAC_Hash reikšmė paskaičiuojama, ir nustatomas ilgis, MAC reikšmė yra įrašoma į WTLS koduojamo teksto struktūrą.

8.1.7.3.7 Saugumo būsena

Ankstesni skyriai detalizavo kaip saugios sesijos yra nustatomos ir sukuriamos. Po nustatymo ir sukūrimo abi pusės turi vienodą saugumo struktūrą, kurioje yra saugumo parametrai, aprašyti sekančioje lentelėje:

2.7 lentelė Saugaus sujungimo saugumo parametrai

Informacija	Paaiškinimas
Sujungimo pusė	Rodo ar esybė yra klientas, ar serveris
Bulk kodavimo algoritmas	Algoritmas naudojamas bulk kodavimui.
MAC algoritmas	Algoritmas naudojamas užtikrinti pranešimų originalumą bei autentiškumo nustatymui.
Suspaudimo algoritmas	Algoritmas naudojamas suspausti duomenis prieš užkodavimą. Visa informacija yra suspaudžiama.
Bendra slapta informacija	20 baitų slapta informacija tarp dviejų saugaus sujungimo pusių
Kliento atsitiktinis skaičius	16 baitų atsitiktinė reikšmė, pateikta kliento
Serverio atsitiktinis skaičius	16 baitų atsitiktinė reikšmė, pateikta kliento
Rakto atnaujinimo dažnis	Laiko intervalas, rodantis kaip dažnai susijungimo būsenos parametrai yra atnaujinami (kodavimo raktas, MAC slapta informacija, pradiniai vektoriai IV)
Eilės numerio metodas	Rodo kuri schema yra naudojama norint sukurti eilės numerius saugiame susijungime. Dabar naudojami metodai yra „*implicit*“, „*explicit*“ ir „*Off*“.

Dabartinė būsena yra nustatoma naudojant saugumo parametrus. Tai reiškia kad dabartinė būsena yra pastoviai atnaujinama. Kiekviena susijungimo būsena savyje turi tokius elementus kaip dabartiniai kodavimo raktai, MAC raktai, pradiniai vektoriai IV (angl. *initial vectors*) bei eilės numeriai. Tiek serveris tiek klientas turi atskirus slaptus raktus kodavimui, MAC, ir pan.

8.1.7.4 WTLS įvertinimas

Žinant saugumo koncepcijas, bei būdus kaip WTLS juos įgyvendina, galima daryti įvertinimą. Keletas saugumo „skylių“ jau yra rastos nepriklausomų tyrėjų, bet sprendimas ar jos yra pakankamai rimtos kad įtakotų visą protokolo modelį, vis dar tiriama. Taigi yra neaišku ar sekančios WTLS versijos sukūrimas galimas be didelių architektūrinių pakeitimų.

8.1.7.4.1 7.4.1. Susiję darbai

SSL yra kuriamas Netscape. Jis yra bendrai priimtas internete autentiškumo nustatymui, ir koduotam bendravimui tarp klientų bei serverių. Internet Engineering Task Force (IETF) organizacijos standartas Transport Layer Security (TLS) yra pagrįstas SSL.

SSL protokolas teikia privatumą, autentiškumo nustatymą, ir originalumą. Duomenys yra užkoduojami naudojant simetrinį kodavimą, o autentiškumo nustatymas vyksta naudojant asimetrinė,

arba viešojo rakto kodavimus. Pranešimų originalumas yra tikrinamas naudojant MAC su raktu. Saugaus hash funkcijos, tokios kaip SHA arba MD5 yra naudojamos MAC skaičiavimams. SSL tikslai yra kriptografinis saugumas, veikimas įvairiose aplinkose, išplečiamumas bei santykinis našumas.

SSL yra sluoksninis protokolas. Kiekviename sluoksnyje pranešimai gali turėti laukus nurodančius ilgį, aprašymą bei turinį. SSL paima perdavimui skirtus pranešimus, sudalina duomenis į apdirbimui tinkančius blokus, gali juos suspausti, pritaiko MAC, užkoduoja ir persiunčia rezultatą. Gauti duomenys yra atkoduojami, patikrinami, atspaudžiami, apjungiami ir pristatomi aukštesnio lygio klientams.

TLS versija 1.0 [19] ir SSL versija 3.0 yra labai panūs standartai, ir skiriasi nedaug, tačiau tarpusavyje yra nesuderinami. Taigi galima teigti jog TSL yra pagerintas ir išplėstas SSL.

8.1.7.4.2 Defektų priežastys

Protokolas buvo sukurtas palaikyti labai platų mobiliųjų įrenginių spektrą. Dauguma silpniausių įrenginių negali palaikyti gero kodavimo dėl jų turimų procesorių, atminties ar srauto pralaidumo apribojimų. Teoriškai, kai klientui leidžiama pasirinkti silpną arba jokio kodavimo, nėra ir apčiuopiamo saugumo. Saugumas negali būti pasiektas įrenginiuose kurie negali vykdyti sudėtingų algoritmų.

Anoniminių prisijungimų įgalinimas yra labai rizikingas. Anoniminis tapatybės nustatymas neretai priveda prie „tarpininko“ (angl. *man-in-the-middle*) tipo atakų. Norint išvengti šios problemos, klientas turėtų apibrėžti kad pasisveikinimas nepalaikys raktų apsikeitimo mechanizmų be autentiškumo nustatymo. Jis visada turėtu nustatyti serverio kurį naudos autentiškumą. Serveris gali prašyti kliento atsiųsti sertifikatą, bet klientas gali atsakyti tuščiu atsakymu. Tada tik nuo serverio priklauso ar jis priims klientą nenustatęs jo autentiškumo. Serverio rakto apsikeitimo pranešimas bus siunčiamas tik kai naudojamas anoniminiai metodai kaip ECDH_anon, RSA_anon ar DH_anon.

Daugelio algoritmų palaikymas padaro sistemą pažeidžiamą, nes kai kurie algoritmai yra „silpni“. Tokiu būdu leidžiant susijungimus naudojančius silpną kodavimą, tik padidina saugumo problemas. Norint išspręsti šią problemą, klientai ir serveriai turėtu naudoti tik stiprius algoritmus.

Taip pat kai kuriose šalyse neleidžiama naudoti algoritmų su pernelyg ilgais raktais. Tokiu būdu negalima teikti apčiuopiamo saugumo. Kita vertus gerai kad WTLS tai numatė, ir dėl to reikalinga tik viena protokolo versija.

8.1.7.4.3 Žinomos saugumo spragos

Keletas potencialių saugumo spragų buvo aptikta WTLS. WTLS specifikacija buvo pritaikyta iš TLS specifikacijos pritaikant kai kuriuos pakeitimus bei modifikacijas. Šios modifikacijos ir pakeitimai bent iš dalies ir privedė prie kai kurių saugumo problemų, tame tarpe ir pasirinktų paprasto

teksto duomenų atstatymo atakos, datagramų nukapojimo atakos, klaidų klastojimo atakos ir rakto paieškos sutrumpinimas kai kuriems eksportuojamiems raktams.

Pradiniai vektoriai, dar vadinami IV, yra naudojami CBC metodo blokų kodavime entropijos sukūrimui. Entropija yra reikalinga norint apsaugoti simetrinį raktą, kuris yra naudojamas CBC metodo blokų kodavime. Be IV, originalus paprastas tekstas būtų koduojamas naudojant pagrindinį (angl. *master*) raktą. Tai atveria galimybę naudoti „neprotingos jėgos“ (angl. *brute force*) metodus to tam kad atrasti „bendros paslapties informaciją“. IV naudojimas apsaugo kad tai neatsitiktų, nes pirmam blokas pakete yra panaudotas XOR su IV. Originalaus paketo žinojimas šiuo atveju nepadėtų, nes jam panaudotas XOR.

Kadangi WTLS turi nepatikimų datagramų palaikymą, kur datagramos gali būti prarandamos, dvejinamos arba keičiama jų eilės tvarka, CBC metodui reikia naujų IV kad užkoduoti kiekvieną paketą. Todėl naudojami IV yra skaičiuojami naudojant XOR su paketo eilės numeriu ir originaliu IV, kuris yra gaunamas rakto generavimo metu. Tai dar yra vadinama linijiniu IV skaičiavimu. Pirmas paprasto teksto blokas pakete yra paveikiamas XOR su paskaičiuotu IV. Originalus IV yra skaičiuojamas reikšmių, siųstų pasisveikinimo metu, pagrindu. Visos reikšmės, įskaitant kliento_atsitiktinę, serverio_atsitiktinę ir eilės numerį yra siunčiami be suspaudimo, taigi jie gali būti nepastebimai perimti. Šie prognozuojami IV priveda prie pasirinkto paprasto-teksto atakų prieš žemos-entropijos bendrą slaptą informaciją. Ši saugumo problema paveikia privatumą.

WTLS palaiko 40 bitų XOR MAC, kuris veikia pranešima papildydamas nuliais, dalindamas į 5 baitų blokus ir tada blokus tarpusavyje paveikdamas XOR. XOR MAC neteikia jokios pranešimo originalumo apsaugos jei srauto kodavimas yra naudojamas, nepriklausomai nuo rakto ilgio. Koduotame tekste bitai gali būti invertuojami, jei invertavimas taip pat daromas ir MAC. Tokiu būdu originalumo patikrinimas rodys kad turinys nebuvo keičiamas, nors jis ir buvo pakeistas. Ši saugumo problema paveikia privatumą.

DES raktas kiekvienam baite turi lygiškumo bitą. Kai naudojamas 40 bitų raktas, efektyvus rakto ilgis naudojamas DES kodavime yra tik 35 bitai (5*7=35). Nepaisant to, a 56 bitų DES raktas turi teisingą kiekį rakto reikmenims, 56 bitų (8*7=56). Geriausias įmanomas saugumo lygis eksporto-nusilpnintuose kodavimo metoduose nebuvo pasiektas. Nors išskleisto raktas turinys 40 bitų DES yra aštuoni baitai, iš tikro rakto efektyvus turinys tik penki baitai. Ši saugumo problema veikia privatumą.

PKCS #1 versijoje 1.5 yra saugumo problema, kai naudojama kartu su protokolu turinčiu patarėją nustatantį ar paketas turi teisingą PKCS #1 versijos 1.5 kamšalą. Jei sistema kokiu nors būdu pasako įsibrovėliui ar naudojamas raktas yra teisingas, sakoma kad sistemoje yra patarėjas. Naudojant šį patarėją įsibrovėlis gali bandyti surasti teisingą raktą, bandant visus galimus ir tikrindamas sistemos atsakymą.

RSA parašai ir kodavimas yra atliekami pagal PKCS #1 versiją 1.5 esančią WTLS, taip leidžiant RSA pranešimus atkoduoti per maždaug 2*e20 pasirinktų kodavimo teksto užklausų. WTLS atveju bad_certificate ir decode_error gali būti patarėju, kurį galima panaudoti nelegaliam atkodavimui. Ši saugumo problema veikia autentiškumo nustatymą.

Neautentikuoti perspėjimų pranešimai, naudojami WTLS, leidžia aktyviam įsibrovėliui pakeisti užkoduotą datagramą neautentikuotu paprasto teksto perspėjimo pranešimu su tuo pačiu eilės numeriu, ir likti nepastebėtu. Ši saugumo problema veikia autentiškumo originalumą (angl. *integrity*).

Laukas record_type yra siunčiamas nekoduotas. Nežinomas stebėtojas gali nustatyti raktų pasikeitimą sekdamas šio lauko turinį. Užkoduoto pranešimo egzistavimas gali būti nustatytas naudojant šį lauką. Ši saugumo problema paveikia privatumą.

„Neprotingos jėgos“ (angl. *brute force*) atakos prieš blokinį kodavimą gali būti panaudojamos, nes teisingi raktai gali visada būti atpažinti bandant atkoduoti kiekvieno paketo paskutinį bloką. Paskutinis blokas yra papildomas iki sekančios pilnų 8 baitų ribos, pildant su pildymo ilgiu.

WTLS specifikacijoje yra iš anksto nurodomos kintamųjų reikšmės Diffie-Hellman skaičiavimams, tačiau dauginimo sub-grupių grupavimo tvarka yra nepaminėta. Dėl to yra neįmanoma patikrinti ar duota vieša reikšmė priklauso teisingai dauginimo sub-grupei. Tai gali būti laikoma smulkia problema, bet ji gali paveikti autentiškumo nustatymą.

40 bitų užkodavimas gali būti lengvai nulaužiamas, todėl taikymai reikalaujantys gero saugumo turėtų neleisti 40 bitų raktų naudojimo. Panašiai, anoniminis Diffie-Hellman yra stipriai nerekomenduojamas, nes jis negali sutrukdyti „tarpininko“ (angl. *man-in-the-middle*) tipo atakų. Pavyzdžiui sertifikatų grandinės su 512 bitų RSA raktais ar signatūromis nėra būdingi didelio saugumo taikymams.

Kai tik serveris nustato autentiškumą, kanalas turėtų tapti vėl atspariu „tarpininko“ (angl. *man-in-the-middle*) tipo atakoms, tačiau visiškai anoniminės sesijos iš prigimties yra pažeidžiamos tokių atakų. Anoniminiai serveriai negali nustatyti klientų autentiškumo, nes jų parašai sertifikato patikrinimo pranešime gali reikalauti kad serverio sertifikatas būtų susietas su tam tikro serverio parašu. Visiškai anoniminiai sujungimai tiekia tik apsaugą nuo pasyvaus nepastebimo informacijos perskaitymo. Jei nepriklausomas ir nesuklastojamas kanalas nėra naudojamas tam kad patvirtinti ar pranešimai nebuvo pakeisti įsibrovėlio, serverio autentiškumo nustatymas yra reikalingas aplinkose, kur „tarpininko“ atakos yra galimos.

Reikia apgalvotai suprojektuoti ir suteikti „pradžios tašką“ (angl. *seed*) pseudo-atsitiktinių skaičių generatoriui PRNG (angl. *Pseudo-Random Number Generator*). Pseudo-atsitiktinių skaičių generatoriai pagrįsti saugiais hash, paprastai MD5 ir / arba SHA, yra priimtini, tačiau jų teikiamas saugumas yra ne didesnis nei galimas atsitiktinių skaičių generatoriaus būsenų kiekis.

512-bitų RSA raktai yra nepakankamai saugūs didelės vertės transakcijoms, ar programoms reikalaujančioms ilgo termino saugumo. Kai viešasis raktas esantis sertifikate negali būti panaudotas užkodavimui, serveris pasitašo laikiną RSA raktą, kurio ir yra pasikeičiama. Laikinas RSA raktas turėtu būti maksimalaus leidžiamo dydžio (t.y., 512 bitų). Raktai turėtų būti dažnai keičiami. Tipinėms elektroninės komercijos taikomosioms programoms, patariama raktus keisti kasdien, arba kas 500 transakcijų, arba jei įmanoma – dar dažniau. Reikia atkreipti dėmesį į tai, kad norint naudoti tą patį laikiną raktą daugiau nei vienai transakcijai, jį reikia kaskart pasirašyti.

„Tarpininko“ (angl. *man-in-the-middle*) tipo užpuolime, užpuolikas turi aktyviai pakeisti vieną ar daugiau pasisveikinimo pranešimų. Jei taip atsitinka, klientas ir serveris suskaičiuos skirtingas reikšmes pasisveikinimo pranešimų abiėse (angl. *hash*). To pasekoje, abi pusės negaus kitų pusių pabaigimo pranešimų. Be „bendros slaptos informacijos" užpuolikas negalės pataisyti pabaigimo pranešimų, ir bus pastebėtas.

8.1.7.4.4 Koks saugumo lygis yra priimtinas?

Saugumo lygio priimtinumas visada yra kompromisas tarp panaudojamumo ir kodavimo metodo gerumo. Neįmanoma nustatyti tam tikrų priimtino saugumo lygio standartų, nes reikalingas saugumo lygis visada priklauso nuo persiunčiamų duomenų. Perduodama informacija visada turi tam tikrą vertę, ir informacijos savininkas turėtų nuspręsti kokio saugumo reikia norint užtikrinti konfidencialumą.

Kompanijos strateginis planas yra ta informacija kuri reikalauja ypatingai gerų užkodavimo metodų. Kita vertus yra informacijos, kuriai nereikia jokio užkodavimo, pavyzdžiui pranešimas, kurio gavėjas yra kviečiamas išgerti puodelį kavos. Tačiau vėlgi – jei yra užkodavimo algoritmai, kodėl gi jų nepanaudojus? Užkodavimas tai daugiau nei tiesiog kodavimo algoritmo pasirinkimas. Skirtingi algoritmai turi skirtingus reikalavimus, norint kad jie sėkmingai veiktų. Paprastai kuo geresnis algoritmas, tuo daugiau skaičiavimo resursų jam reikia.

WTLS atveju teikiamas saugumo lygis visada yra apribotas turimų ribotų resursų. Nėra prasmės naudoti daugiau nei 50% turimų, ir taip jau labai ribotų skaičiavimo resursų užkodavimui ir atkodavimui, bei sumažinti ir taip jau mažą pralaidumą. Tačiau WTLS turi užtikrinti tam tikrą saugumo lygį, tam kad jį būtų galima naudoti komerciniams tikslams.

Didžiausias WTLS trūkumas yra tas, kad vartotojui leidžiama pasirinkti itin silpnus algoritmus. Silpno algoritmo pavyzdžiu galime vadinti SHA_XOR_40, kuris turėtų užtikrinti duomenų originalumą (angl. *integrity*). Kitas trūkumas yra tas, kad nors serveris ir renkasi naudojamą algoritmą, jis renkasi iš sąrašo pateikto kliento, o klientas netgi gali nurodyti labiausiai pageidaujamą. Tiesa jei klientas nepasiūlo algoritmų kuriuos serveris sutiktų naudoti, susijungimas gali būti nesukuriamas.

Žemiau lentelėje pateikiami skaičiai, rodantys apytikslius duomenis, kiek laiko ir piniginių išlaidų reikėjo norint įvykdyti aparatūrines „neprotingas“ (angl. *brute force*) atakas 1999 metais.

2.8 lentelė Vidutiniai laiko įverčiai vykdant aparatūrines „neprotingas“ atakas

	Rakto ilgis bitais					
Kaina [$]	**40**	**56**	**64**	**80**	**112**	**128**
100 K	2 s	35 val	1 met	70000 met	10^{14} met	10^{19} met
1 M	0.2 s	3.5 val	37 d	7000 met	10^{13} met	10^{18} met
10 M	0.02 s	21 min	4 d	700 met	10^{12} met	10^{17} met
100 M	2 ms	2 min	9 h	70 met	10^{11} met	10^{16} met
1 G	0.02 ms	13 s	1 h	7 met	10^{10} met	10^{15} met

Atsižvelgiant į Mūro dėsnį (angl. *Moore's Law*), bei kitas aplinkybes, šiandienos įverčiai gali būti gauti padalinus skaičius iš 15. WTLS atveju dauguma užkodavimo algoritmų naudojamų komunikavimo kanalo kodavimui bus RC5_CBC su 40 ir 56 bitų raktais ir DES_CBC su 40 bitų raktu. Nereikia net sakyti kad 40 ar 56 bitų raktų neužtenka. Nėra techninių apribojimų neleidžiančių naudoti ilgesnius raktus, netgi dabartiniai turimi procesoriai ir pralaidumas yra pakankami geresniam užkodavimui. Tiesa WTLS palaiko ir 3DES_CBC_EDE su 168 bitų raktu.

Be trumpo rakto ilgio, yra ir kitų priežasčių trūkumams. Pavyzdžiui DES grįstų algoritmų atveju, dalis rakto bitų yra naudojama lygiškumo nustatymui. Tai reiškia kad efektyvus rakto ilgis gali būti mažesnis. Pavyzdžiui DES_40 realus naudingas saugumui rakto ilgis yra 35 bitai, o DES_64 atveju realus naudingas saugumui rakto ilgis yra 56 bitai. Bendru atveju – DES yra senas standartas, ir yra keletas būdų jam įveikti.

Viešieji raktai yra naudojami pasikeičiant „pradine bendra slapta informacija“, tačiau svarbu kad tai būtų daroma saugiai. Sekanti lentelė rodo rekomenduojamus viešųjų raktų ilgius:

2.9 lentelė Rekomenduojami viešųjų raktų ilgiai (bitais)

Metai	**Individualiam naudojimui**	**Įmonių naudojimui**	**Valstybinėms organizacijoms**
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

WTLS atveju, anoniminiam raktų apsikeitimui yra galimos tik 512 ir 768 bitų versijos. Sertifikuotam autentiškumo nustatymui iš WTLS pusės apribojimų nėra. Priimant prielaidą kad sertifikuotam autentiškumo nustatymui naudojami raktų ilgiai virš 1024 bitų, WTLS teikia pakankamą saugumo lygį konfidencialiam pasikeitimui raktais ir autentiškumo nustatymui. Tačiau kai kalbama

apie anonimišką pasikeitimą raktais, 512 bitų raktas yra pernelyg trumpas tam kad raktų apsikeitimas būtų saugus. 768 bitų raktas yra ant priimtino saugumo ribos.

MAC su raktu funkcionalumas, SHA ir MD5, teikiamos WTLD gali būti laikomos saugiomis, jei pilnas rakto ilgis yra naudojamas. Šiuo metu nėra žinomų kriptografinių atakų prieš SHA ar MD5. Tačiau sakoma kad MD5 turi silpnybę suspaudimo funkcijoje, nors praktinės įtakos šios hash funkcijos saugumui tai nedaro.

8.1.7.5 Išvados

WTLS yra pirmas bandymas teikti saugų baigtis-baigtis susijungimą belaidei taikymų aplinkai WAE (angl. *Wireless Application Protocol*). Dažnai naudojami protokolai, tokie kaip TLS versija 1.0 ir SSL versija 3.0 buvo pritaikyti kaip WTLS pagrindas. Tačiau nebuvo galima pritaikyti visų procedūrų, naudojamų tradicinėje susijungimais grįstoje aplinkoje. Kūrybiniai darbai pasibaigė protokolu, kuris yra panašus į TLS, tačiau turi kai kurių savybių, leidžiančių jį pritaikyti belaidžiui tinklui.

WTLS palaiko gan plačią algoritmų aibę, tam kad atitikti privatumo, autentiškumo nustatymo ir duomenų originalumo reikalavimus. Šiuo metu privatumas yra įgyvendintas naudojant blokų kodavimą, tokį kaip DES_CBC, IDEA, ir RC5_CBC. RSA bei Diffie-Hellman raktų apsikeitimo rinkiniai yra palaikomi komunikuojančių pusių autentiškumo nustatymui. O duomenų originalumas yra įgyvendintas panaudojant SHA-1 ir MD5 MAC algoritmus.

Prieita prie išvados jog galima garantuoti pakankamą saugumo lygį, su ta sąlyga, jog bus naudojamos tam tikros palaikomų algoritmų kombinacijos. Tokiu būdu galima išvengti daugumos žinomų saugumo trūkumų. Pavyzdžiui naudojant gerą autentiškumo nustatymą, RSA raktų apsikeitimą su RSA grįstais sertifikatais, raktus ne mažesnius nei 1024 bitai, blokų kodavimą, RC5 su 56 bitų raktu ir pilną MAC algoritmą, SHA-1, eilinis vartotojas gali būti pakankamai tikras saugumo lygių. Tiesa WTLS leidžia klientui ar serveriui pasirinkti nulinio kodavimo algoritmą, tai reiškia komunikavimą neužtikrinant privatumo, autentiškumo nustatymo ir duomenų originalumo reikalavimų. Dar daugiau – pasirinkti algoritmai turi būti subalansuoti – nėra prasmės naudoti 512 bitų viešųjų raktų apsikeitimo algoritmą jei blokų kodavimas bus tik 168 bitų.

Taigi priėjome išvadų jog WTLS nėra jokių didelių saugumo spragų, todėl didelių architektūrinių pakeitimų neprireiks. Tačiau kai kurie technologiniai pakeitimai ir nedideli procedūriniai pakeitimai yra reikalingi. Technologinio pakeitimo pavyzdžiu galėtų būti dabartinės PKCS#1 versijos 1.5 pakeitimas į versiją 2.0, kurioje jau ištaisyta keletas saugumo spragų. RSA yra naudojamas tiek autentiškumo nustatymui, tiek ir duomenų originalumo nustatymui WTLS. Kitas taisytinas defektas, yra pradiniai vektoriai IV (angl. *initial vectors*) blokų kodavime. Visos skaičiavimui reikalingos reikšmės yra žinomos besiklausančiai pusei. Tam tikra unikali ir slapta

informacija turėtų būti naudojama IV skaičiavimams, norint suteikti daugiau saugumo. Tačiau slapta informacija turėtų būti ne ta pati, kuri yra naudojama kaip kodavimo raktas blokų kodavime.

Įrašų sluoksnis naudoja papildomą eiliškumo numeravimą. Nepaisant to, eiliškumo numeravimas datagramų transportavime yra būtinas. Pastebima, jog naudojamas eilės numeris yra perduodamas be užkodavimo. Slaptai perimančiam pranešimus tai padeda sugeneruoti naudojamus pradinius vektorius. Sekančiose WTLS versijose reikėtų šią savybę pakeisti taip, kad eilės numeris būtų koduojamas.

Tam kad išvengti „tarpininko“ (angl. *man-in-the-middle*) tipo atakų, anoniminis autentiškumo nustatymas turėtų būti uždraustas, bent jau serverio pusėje. Viena prioritetinių sričių yra viešojo rakto infrastruktūroje, ypač belaidėje aplinkoje. Problema yra patikima „trečioji pusė“ (angl. *third parties*). Kas ja bus? Ar mobilieji vartotojai nori gauti sertifikatus iš dabartinių sertifikatų tiekėjų, ar iš operatorių, kurių klientų identifikatorius jie turi? Kaip klientai, vartotojai nori sertifikatus gauti kuo lengviau.

Padarėme išvadą jog WTLS negali suteikti privatumo, kurį būtų galima taikyti vertingai informacijai, tol kol nenaudojami geri blokų kodavimo mechanizmai. 40 bitų raktas negali būti laikomas pakankamu jokiomis aplinkybėmis, o 56 bitų raktas yra tiek priimtinumo riba. Tačiau, kaip minėta anksčiau, patikimas autentiškumo nustatymas ir garantuotas duomenų originalumas kasdieniniame naudojime yra svarbiau nei privatumas. Slaptesnei informacijai reikėtų naudoti bent 128 bitus. Ateityje į WTLS bus įtrauktas ir srautų (angl. *streams*) kodavimas – lieka tik palaukti ir pamatyti ar jie išspręs esamas saugumo problemas, ar sukurs naujas.

Nepaisant SSL ir TLS defektų, jie įgavo apčiuopiamą populiarumą reikiant saugumą susijungimais orientuotame komunikavime. Yra laikoma kad jie tiekia pakankamą saugumo lygį. Panašiai galima vertinti ir WTLS. Jis nėra tobulas, tačiau yra pakankamas protokolas. Tam kad padėti WAP plisti, žmonės turi pasitikėti WTLS tiekiamu saugumu. Vartotojai turi visada žinoti jog absoliutus saugumas yra neįmanomas. Bet kokiomis aplinkybėmis kai vartotojai naudoja paskirstytas paslaugas tuo pačiu ir tam tikru laipsniu rizikuoja. Šį, su tinkliniu komunikavimu susijusį rizikos laipsnį jie turi priimti kaip būtiną, ir tada nuspręsti kokio svarbumo duomenis jie gali perdavinėti. Tokios pačios taisyklės galioja ir WTLS bei WAP.

8.1.8 Belaidžių sesijų protokolas WSP

Belaidžiame sesijų protokole WSP (angl. *Wireless Session Protocol*), belaidė sesija reiškia pasaulinio žiniatinklio naršymo sesiją. Sesija prasideda kai vartotojas prisijungia prie vieno URL. Sesija baigiasi tik vartotojui palikus tą URL. WSP protokolas nenurodo, kad tai yra vienintelė jo paskirtis, bet visas protokolas yra suprojektuotas būtent tam..

Sesija: Ilgai išliekantis komunikavimo kontekstas, įkurtas tarp dviejų programų, transakcijų ir tipizuotų duomenų perdavimo tikslais.

Įkuriant sesiją, sesijos srities nustatymai turi būti nustatyti kartą, pačioje sesijos pradžioje. Tai padeda sumažinti pralaidumo naudojimą, t.y. reikia siųsti mažiau duomenų. Dėl belaidžio komunikavimo prigimties, sesijų įkūrimo procesas turi būti efektyvus, o ilgas pasisveikinimas (angl. *hand shaking*)– nepageidaujamas.

WSP yra pagrįstas HTTP 1.1, su keliais pagerinimais. WSP tiekia aukštesnio lygio WAP taikymų sluoksniui pastovią sąsają dviems sesijų paslaugoms. Pirma yra susijungimais-orientuota paslauga, kuri dirba virš transakcijų sluoksnio protokolo WTP. Antroji yra ne susijungimais orientuota paslauga, kuri dirba virš saugios arba nesaugios datagramų transportavimo paslaugos. Taigi WSP egzistuoja dėl dviejų priežasčių – pirma, susijungimais-orientuotame rėžime jis pagerina HTTP 1.1 našumą per belaidžius tinklus. Antra, jis teikia sesijų sluoksnį taip, kad visa WAP aplinka tampa panaši į ISO OSI informacinį modelį.

8.1.8.1 WSP tarpsluoksninio komunikavimo elementai

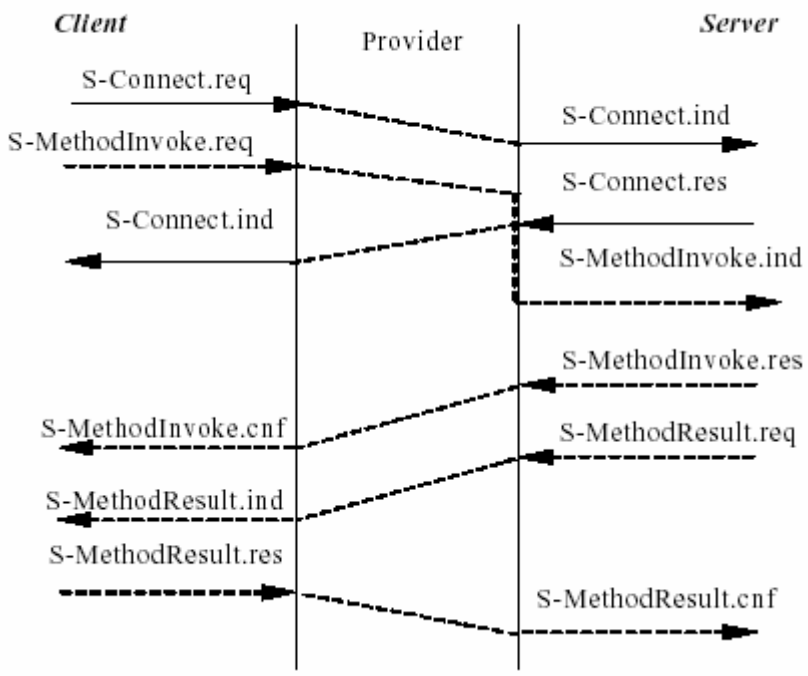
8.1.8.1.1 Susijungimais orientuotos sesijos paslauga

Susijungimais-orientuotos paslaugos rėžime naudojamos visos WSP galimybės:

1. Visi paslaugos primityvai
2. Reikalaujama WTP naudojimo
3. Teikia sesijų paslaugas su tam tikru HTTP funkcionalumu.

Susijungimais-orientuotas WSP palaiko šias savybes:

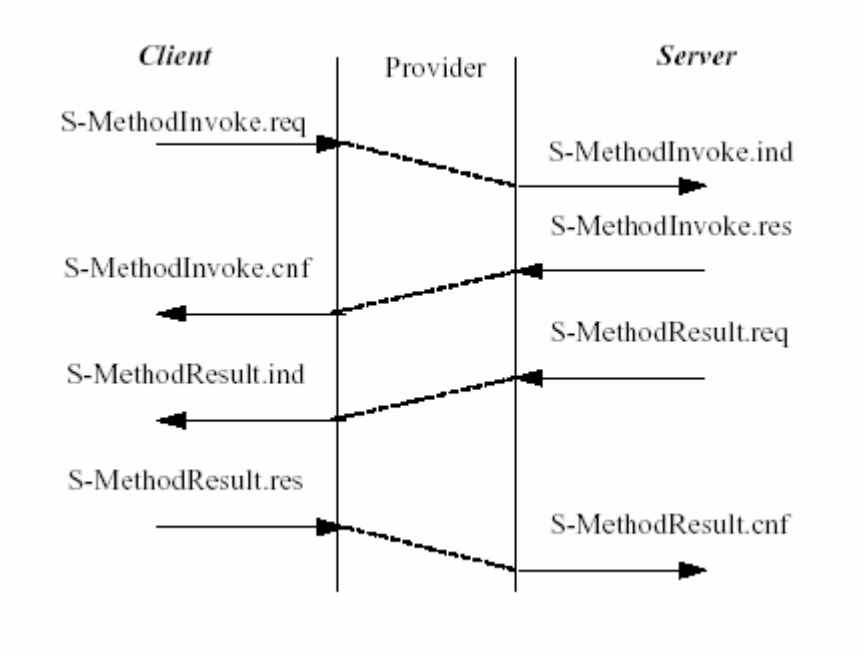
- sesijos įkūrimą
- metodų iškvietimą / užklausas
- *push* pranešimus
- laikiną sustabdymą (angl. *suspend*)
- atnaujinimą (angl. *resume*)
- sesijos užbaigimą



Sesijos įkūrimo pavyzdys

Galimybių suderinimas vyksta sesijos įkūrimo metu. Tik vienos pusės galimybių suderinimas yra apibrėžtas, kuriame iniciatorius pasiūlo galimybių aibę, o gavėjas atsako. Kadangi WAP serveris yra daug galingesnis nei WAP įrenginys, tikimasi kad serveris galės atitikti visas įrenginio pasiūlytas galimybes.

Žemiau schemoje pateiktas metodų iškvietimo pavyzdys - WSP metodai atitinka HTTP 1.1 metodus.



WSP metodų iškvietimo pavyzdys

8.1.8.1.2 Ne susijungimais orientuotos sesijos paslauga

Ne susijungimais-orientuotos paslaugos rėžime naudojamos visos WSP galimybės:

1. Tik paprastas užklausas - reply ir *push*, nėra sesijos valdymo
2. Nepriklauso nuo WTP
3. teikia paslaugas panašias į „*Smart Messaging*" sesijas

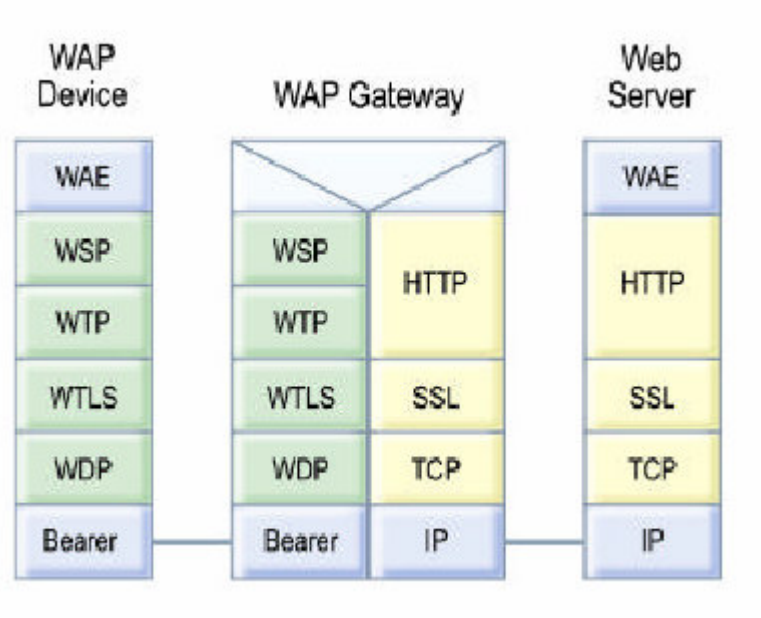
Ne susijungimais-orientuotas WSP palaiko tik šias tris savybes:

1. *MethodInvoke* – užklausos siuntimas
2. *MethodResult* – atsakyti į užklausą
3. *UnitPush* – „*push*“ pranešimo siuntimas

Kiekvienas „*push*“ pranešimas arba „*invoke*“ / „*result*“ pora yra identifikuojama naudojant TID. Nėra tikros protokolo logikos. Komunikavimas galimas bet kuriuo laiko momentu, jei tinklas yra prieinamas. Saugumo paslaugos (WTLS) gali būti naudojamos. Kadangi nenaudojamas WTP ne susijungimais-orientuotas WSP gali būti nepatikimas!

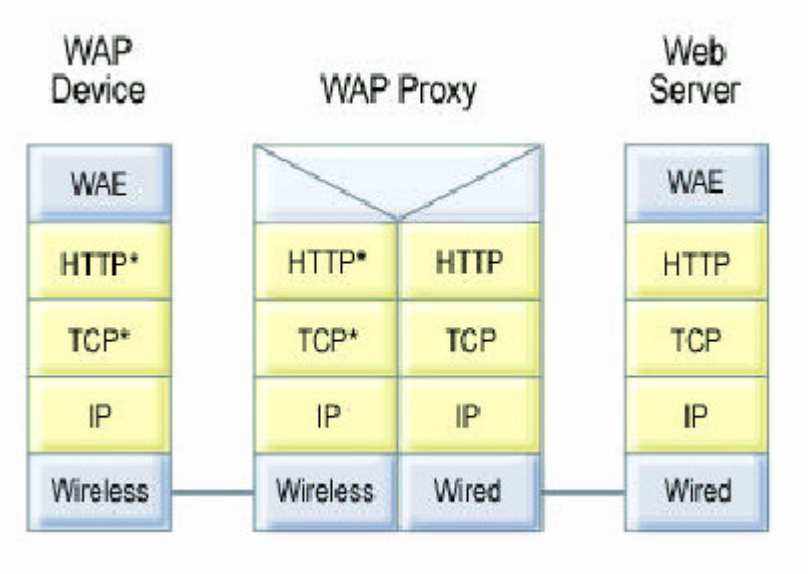
8.1.9 Pavyzdinės WAP technologijos konfigūracijos

Kadangi keletas WAP dėklo paslaugų gali būti tiekiamos naudojant skirtingus protokolus, priklausomus nuo aplinkybių, yra daugiau nei viena galima dėklo konfigūracija. Sekančios schemos iliustruoja keletą įmanomų protokolo dėklų naudojant WAP technologiją.



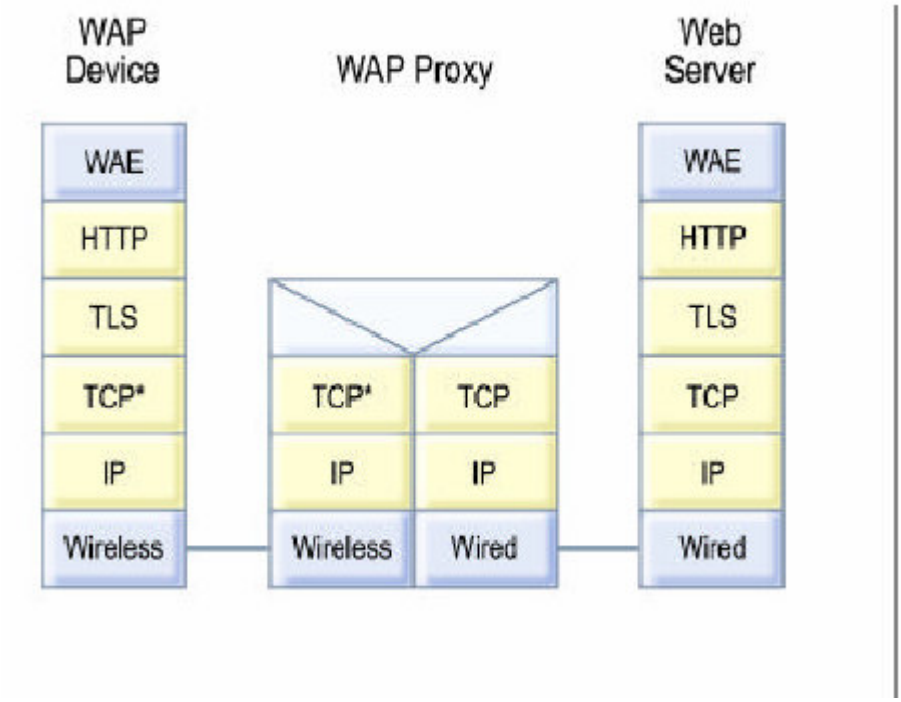
WAP 1.x vartų (angl. *gateway*) pavyzdys

Ši schema iliustruoja protokolų dėklą originaliai WAP architektūrai. WAP vartai perverčia hiperdaugialypės terpės transportavimo paslaugą tarp datagramomis grįstų protokolų (WSP, WTP, WTLS, WDP) ir susijungimais-orientuotų protokolų, dažnai naudojamų internete (HTTP, SSL, TCP).



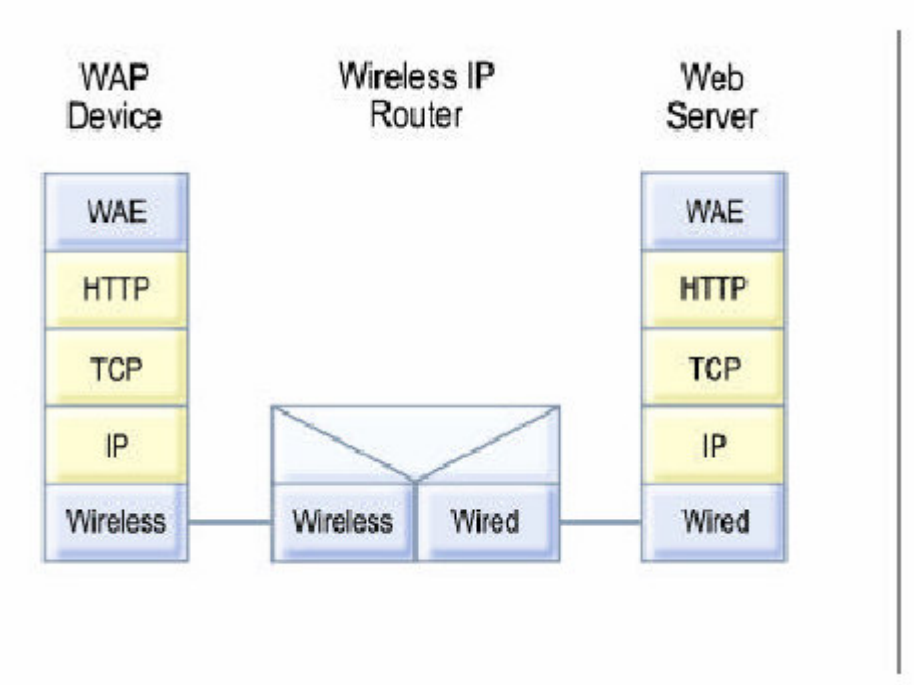
WAP HTTP tarpinio serverio (angl. *proxy*) su TCP ir HTTP pavyzdys.

Šioje schemoje iliustruojamas WAP HTTP tarpinis serveris. Tarpinio serverio konfigūracija yra plačiai naudojama internete norint pasiekti įprastinio žiniatinklio ar daugialypės terpės duomenis – muziką, vaizdo medžiagą ir pan. Ši konfigūracija „pastato“ WAP tarpinį serverį tarp laidinio ir belaidžio tinklų, tam kad pagerinti belaidžio TCP (pavaizduoto kaip TCP*) profilio našumą. Be TCP optimizacijų, pagerintas belaidžio HTTP (pavaizduoto kaip HTTP*) profilis leidžia tolesnius našumo pagerinimus. Abu profiliai susideda iš gerai apibrėžtų IETF nustatymų, kurie leidžia efektyvų veikimą belaidžiais WAP palaikomais tinklais. Todėl belaidės profiliuotos versijos gali veikti su TCP ir HTTP.



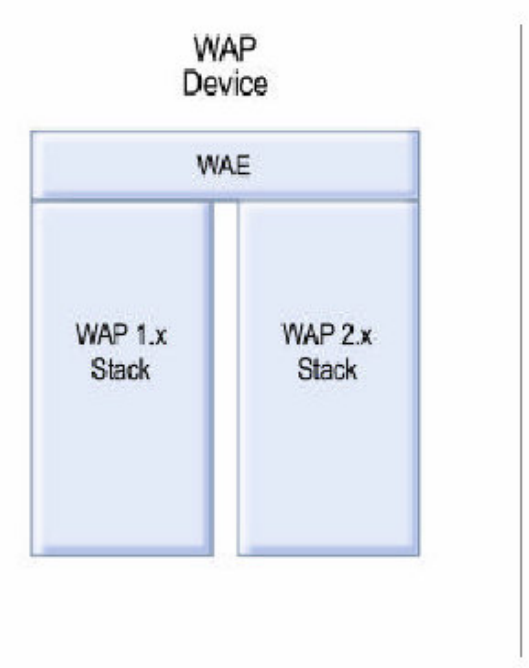
WAP tarpinio serverio TLS tunelio palaikymo pavyzdys

Ši iliustracija rodo WAP HTTP tarpinio serverio susijungimu-grįstą tunelį į žiniatinklio serverį (pvz. atsakant į susijungimo komandą *CONNECT*). Ši konfigūracija yra naudojama leisti TLS teikti baigtis-baigtis (angl. *end-to-end*) saugumą mobiliesiems terminalams ir kilmės serveriui. Elektroninė komercija yra dažnas baigties-baigties saugumo pavyzdys.



Tiesioginio sujungimo pavyzdys

Ši iliustracija rodo WAP įrenginio tiesioginį prisijungimą prie pasaulinio žiniatinklio serverio internete. Belaidis IP maršrutizatorius yra standartinė IP tinklo dalis, kuri yra skirta perduoti IP paketus iš vieno susijungimo sluoksnio (pvz. belaidžio tinklo) į kitą (pvz. laidinį tinklą). Ši konfigūracija gali būti pritaikoma ir ten, kur yra naudojamas nešančiojo sluoksnio saugumas (pvz. toks kaip IPSec). Tiesioginio sujungimo atveju belaidės optimizacijos apibrėžtos belaidžių profilių (angl. *Wireless Profiles*) TCP ir HTTP gali būti nepritaikomos.



Dvigubo dėklo palaikymas

Paskutinė iliustracija rodo atskirus protokolo dėklus, naudojamus kiekvienai WAP konfigūracijai – šis įrenginys palaiko tiek WAP 1.x tiek ir 2.x protokolų dėklus. Ši konfigūracija yra naudinga tais atvejais, kai įrenginys turi bendrauti tiek su naujais tiek su senais WAP serveriais.

8.1.10 LEAP – alternatyva WAP

Lengvas ir efektyvus taikymų protokolas LEAP (angl. *Lightweight and Efficient Application Protocol*) yra bendras karkasas didelio našumo efektyvių protokolų, kurie yra idealūs belaidžiui bei mobiliam taikymui, aibei. LEAP yra suprojektuotas atsižvelgiant į visus belaidės duomenų komunikacijos industrijos reikalavimus, ir yra orientuotas į didžiausios naudos industrijai bei vartotojui teikimą.

LEAP protokolų patentai nekainuoja, o atviro kodo protokolų įgyvendinimai jau yra sukurti įvairiems įrenginiams bei pranešimų centrų platformoms. Kadangi protokolai yra sukurti ir prieinami, bei gali būti greitai išplatinti bei įgyvendinti kaip perspektyvi WAP alternatyva.

8.1.10.1 Trumpa LEAP istorija

LEAP atsirado 1994, kaip McCaw Cellular (dabar AT&T Wireless) grupės tyrimų bei įgyvendinimo iniciatyva. Tuo metu McCaw Cellular buvo susikoncentravusi ties pranešimų perdavimo (angl. *paging*) paslauga ir buvo neseniai nusipirki belaižio siaurajuosčio ryšio PCS licenzijas (JAV), todėl norėjo sukurti efektyvią belaidžių pranešimų transportavimo bei pristatymo sistemą. Neda Communications, Inc., nepriklausoma konsultacijų įmonė, dirbusi McCaw Cellular pagal kontraktą, atliko svarbų vaidmenį šios sistemos įgyvendinime. Neda Communications taip pat nuo pat pradžių buvo ir CDPD specifikacijos kūrime.

Tačiau po to kai 1997 metais McCaw Cellular perpirko AT&T, kompanija paliko siaurajuosčio ryšio PCS pranešimų perdavimo rinką. Prieš tam atsitinkant, Neda Communications apsaugojo nuo AT&T nepriklausomam protokolų kūrimui reikalingas teises. Tačiau suprasdama šių protokolų reikalingumą netolimoje ateityje, Neda apsiėmė ir toliau kurti šiuos protokolus, nepriklausomai nuo AT&T. Netrukus Neda juos užbaigė, paskelbė kaip RFC, ir dabar jie yra LEAP protokolų pagrindas.

8.1.10.2 Techninė LEAP apžvalga

Šiame skyriuje padarysime trumpą techninę LEAP protokolų apžvalgą. Norint gauti detalesnės informacijos apie LEAP protokolus, jos reikėtų ieškoti „*The LEAP Manifesto“ kuris yra čia* - http://www.freeProtocols.org/leap.

LEAP yra belaidžio taikymo protokolų aibė, optimizuota mažų pranešimų perdavimui belaidžiais tinklais. Belaidžiai tinklai yra ribojami nedidelio pralaidumo, o nedidelių gabaritų įrenginiai yra ribojami tokių dalykų kaip ekrano dydis, baterijų bei atmintinės talpa. Šie apribojimai labai sureikšmina duomenų perdavimo efektyvumą.

LEAP protokolai yra apie penkis kartus efektyvesni nei įprastas SMTP elektroninių laiškų protokolas. Šis pagerintas efektyvumas reiškia ilgesnį baterijų darbo laiką mobiliesiems telefonams, delniniams kompiuteriams bei kitiems belaidžio interneto įrenginiams,

8.1.10.2.1 LEAP sluoksniai

LEAP protokolai yra sluoksniuoti. Žemesnysis sluoksnis, vadinamas „*Efficient Short Remote Operation*“ (ESRO), teikia patikimas, ne susijungimais orientuotas paslaugas, kurios gali būti naudojamos įvairiais tikslais. Pavyzdžiui, be mobiliųjų pranešimų paslaugų, ESRO taip pat gali būti naudojamas kaip transportas kreditinių kortelių tikrinimo programoms, bei efektyvioms mikro-naršyklėms. Virš ERSO yra sluoksnis vadinamas EMSD. EMSD yra pranešimų protokolas kuris yra gerai optimizuotas trumpų interneto pašto pranešimų siuntimui ir gavimui.

8.1.10.2.2 ESRO

Visos efektyvios programos turi efektyvaus transportavimo mechanizmo poreikį. Dėl to daugiausia dėmesio kuriant bei įgyvendinant protokolą buvo skirta būtent bendro efektyvaus transportavimo mechanizmo kūrimui. To pasekoje ir atsirado kaip tik toks protokolas – ESRO (angl. *Efficient Short Remote Operations*). ESRO yra patikimas, ne susijungimais orientuotas transportavimo mechanizmas, suformuojantis efektyvių protokolų kūrimo pagrindą kai TCP yra perdaug o UDP yra per mažai.

8.1.10.2.3 EMSD

ESMD (angl. *Efficient Mail Submission and Delivery*) yra protokolas sukurtas „ant“ ESRO, ir yra skirtas efektyviam mobiliųjų pranešimų siuntimui ir gavimui.

8.1.10.2.4 Pagrindinis tikslas: mobilieji pranešimai

Efektyvių protokolų poreikis tęsiasi per visus belaidžio komunikavimo aspektus – pradedant elektroniniu paštu, baigiant pasaulinio žiniatinklio naršymu bei kitais taikymais. LEAP architektūra priderinta prie visų šių taikymų. Tiesa pagrindiniai LEAP protokolai yra skirti palaikyti mobiliųjų pranešimų taikymus, nes tai yra dominuojantis taikymas bevieliuose didelės apimties tinkluose.

8.1.10.3 Procesai ir procedūros

8.1.10.3.1 RFC publikavimas

Abu protokolai buvo publikuoti kaip interneto RFC. ESRO yra RFC-2188, o EMSD yra RFC-2524. RFC publikavimas yra vyraujanti publikavimo procedūra internete, užtikrinanti kad protokolai yra nemokamai, lengvai ir ilgam prienami visiems kas nori jais pasinaudoti.

8.1.10.3.2 Patentai

Kaip aptarta „*The WAP Trap*“ (http://www.freeprotocols.org/wapTrap/), labai norima protokolo standarto savybė yra patentų nebuvimas. Patentuotų komponentų buvimas protokole kenkia pagrindiniam protokolo tikslui: neribotam jo pritaikymui ir naudojimui.

Neda paskelbė atvirų protokolų fondui (angl. *Free Protocols Foundation*) kad LEAP protokolai yra be patentų, ir kad jie tokie bus visą laiką.

8.1.10.3.3 Atviros priežiūros organizacijos

Norint suteikti atvirą forumą LEAP protokolų tolesniam plėtojimui ir priežiūrai, Neda įkūrė atskiras viešąsias organizacijas kiekvienam protokolui.

Todėl ESRO ir EMSD protokolai yra prižiūrimi ir palaikomi atitinkamai ESRO.org adresu http://www.esro.org/, bei EMSD.org adresu http://www.emsd.org/.

Kiekviena iš organizacijų leidžia atvirą atitinkamo protokolo peržiūrą, bei suteikia mechanizmus jiems taisyti bei tobulinti.

Bet kuris suinteresuotas asmuo gali tapti šių organizacijų nariu, bei dalyvauti tolimesniame protokolų kūrime. Dalyvavimas kūrimo procese yra visiškai atviras, ir nereikalaujantis privilegijų; taip pat nėra ir nario mokesčio. Vienintelis reikalavimas yra kad dalyviai turi laikytis atvirų protokolų fondo (angl. *Free Protocols Foundation*) principų bei procedūrų, tuo pačiu užtikrinant kad protokolai išliks be patentų.

8.1.10.4 LEAP ir WAP palyginimas

„*The WAP Trap*“ (http://www.freeprotocols.org/wapTrap/) yra įvardinti pagrindiniai skirtumai tarp WAP ir LEAP. Žinoma reikia atsižvelgti į tai, jog „*The WAP Trap*“ *yra* atvirų protokolų fondo (angl. *Free Protocols Foundation*) kuris rūpinasi LEAP ateitimi, svetainėje. Sekančioje lentelėje yra įvertintos pagrindinės bendros WAP ir LEAP charakteristikos.

2.10 lentelė WAP ir LEAP palyginimas

WAP	LEAP
Reikia žinoti patentų apribojimus	Nėra patentų
Publikuotas savarankiškai, WAP Forum	Publikuotas kaip RFC
Versijos gali keistis be perspėjimo	Visos versijos yra fiksuotos
Prižiūrimas WAP Forum	Prižiūrimas atviros darbo grupės
Egzistuojančių protokolų „atradimas iš naujo“	Efektyvumą optimizuojantys išplėtimai egzistuojantiems protokolams
Priklausomas nuo įrenginio vartotojo sąsajos charakteristikų	Nepriklausomas nuo vartotojo sąsajos
Būdingos saugumo spragos	Nėra saugumo prielaidų
Nepastovus protokolo numerio priskyrimas	Pastovus protokolo numerio priskyrimas
Prastas techninis modelis	Geras techninis modelis
Pagrindinis tikslas: žiniatinklio naršymas	Pagrindinis tikslas: pranešimų siuntimas

8.1.10.4.1 Patentų ribojimas

Kaip pažymima „*The WAP Trap*“, WAP specifikacijose yra patentuotų komponentų. Priešingai nei WAP – protokolai yra visiškai be patentų.

8.1.10.4.2 Publikavimo atvirumas

Kaip rašyta ankstesniuose skyriuose, LEAP protokolai yra paskelbti kaip internet RFC, užtikrinant pastovų, neribotą protokolų buvimą. Kita vertus WAP specifikacijos yra savarankiškai paskelbtos WAP Forum, o tai negarantuoja tokio neriboto prieinamumo. WAP specifikacijos prieinamumas ir našumas yra tiek pat geras kiek ir pati WAP Forum organizacija.

Tiriant toliau matome jog norint parsisiųsti bet kurią WAP specifikaciją, vartotojas turi sutikti su licenzijos sutartimi. Tuo tarpu LEAP protokolai gali būti parsisiunčiami bei platinami be jokio licenzijos ribojimo.

Taip pat WAP Forum organizacijos publikavimo filosofija negarantuoja stabilumo. Pradedant 2000 metų vasario mėnesiu, WAP specifikacijos tituliniame puslapyje yra užrašas skelbiantis jog šis dokumentas gali keistis be jokio perspėjimo (angl. „*This document is subject to change without notice*“). Tuo tarpu RFC publikavimo procesas užtikrina tai, kad kiekviena LEAP protokolo versija yra visam laikui nekintanti.

8.1.10.4.3 Priežiūros atvirumas

LEAP atviras priežiūros procesas taip pat yra didelis kontrastas WAP. Dalyvavimas WAP specifikacijų kūrime (nuo 2000 metų vasario mėnesio) reikalauja WAP Forum nario mokesčio - 27,000 JAV dolerių, bei vyksta už „uždarų durų“. Priešingai nei WAP, LEAP protokolai yra prižiūrimi organizacijų, kuriose visi gali dalyvauti nemokamai.

8.1.10.4.4 Techniniai trūkumai

WAP protokole taip pat yra ne vienas techninis trūkumas. Kaip buvo svarstoma „*The WAP Trap“* (informacija gali būti tendencinga), WAP yra plataus masto egzistuojančių protokolų atradimas iš naujo. Tuo tarpu LEAP protokolai susideda iš nedidelio kiekio nepriklausomų protokolų, kurie papildo esamus interneto protokolus.

8.1.10.4.5 Pagrindinis tikslas

Tarp WAP ir LEAD yra didelių koncepcinių skirtumų, kuriuos čia aprašysime. Visų pirma LEAP yra orientuotas į mobiliųjų pranešimų (pvz. elektroninio pašto) taikymus, kai WAP yra orientuotas į mobiliojo žiniatinklio naršymą. „*The WAP Trap“* nuomone tai rodo didelį mobiliojo komunikavimo industrijos nesupratimą iš WAP Forum pusės. Tuo tarpu „*The WAP Trap“* nuomone mažų gabaritų įrenginiai yra puikiai tinkantys elektroniniam paštui, tačiau dėl vartotojo sąsajos apribojimų, visiškai netinkami žiniatinklio naršymui.

Taip pat LEAP bei WAP labai skirtingai žiūri į žinučių perdavimą. Šie du požiūriai yra vienas kitą papildantys. LEAP požiūris, kuris yra įkūnytas EMSD protokole, yra pilnas ir efektyvus siuntimo ir gavimo modelis. WAP požiūris yra tiesiog priėjimas prie pašto dėžutės, ir pasirinktų pranešimų parsisiuntimas.

8.1.10.5 LEAP išplitimas

Nors teoriškai LEAP turi daug privalumų lyginant su WAP (tai itin afišuoja „*The WAP Trap“*), LEAP nėra toks paplitęs kaip WAP, o jo ateitis atrodo miglota.

Tam kad LEAP protokolai būtų plačiai naudojami, jie turi būti įgyvendinti programinių sprendimų forma, kuriuos galėtų naudoti galutiniai vartotojai. Tam kad tai pasiekti, Neda sukūrė atviro kodo programinius protokolų įgyvendinimus daugumai platformų. Protokolų varikliai yra prieinami pernešamo programos kodo pavidalu, kuris buvo nukeltas į įvairias platformas. Žiūrint iš įrenginių pusės, programinė įranga yra prieinama pranešimų gavėjams, mobiliesiems telefonams, delniniams kompiuteriams (Windows CE, Palm OS, Palm PC, EPOC, Palm Pilot), asmeniniams kompiuteriams (Windows, UNIX, DOS). Pranešimų centrų pusei įranga yra prieinama Windows, Solaris, bei Linux sistemose.

8.1.10.6 Kitos alternatyvos WAP

Šiuo metu kitų žinomų bei apčiuopiamus rezultatus pasiekusių alternatyvų WAP nėra.

8.1.11 Išvados

Apie WAP galima pasakyti jog tai yra dar palyginus jauna technologija, kurios trūkumai yra sparčiai šalinami. WAP įgauna vis daugiau funkcionalumo, išbaigtumo ir saugumo.

WAP apibrėžia alternatyvą pasauliniam žiniatinkliui, turinio nešantiesiems tinklams. WAP modelis buvo optimizuotas atsižvelgiant į apribojimus, kurie yra labai skirtingi lyginant su šiandienos žiniatinkliui taikomų apribojimų:

- Belaidžiai įrenginiai, tokie kaip mobilieji telefonai, turi palyginus mažus ekranus
- Belaidžiai įrenginiai paprastai neturi pelės ir turi ribotą klaviatūrą, kuri paprastai yra naudojama viena ranka
- Belaidžiai įrenginiai paprastai turi mažai apdorojimo galios ir atmintinės.
- Belaidžiai tinklai turi polinkį į ryšio praradimą, arba jo suprastėjimą
- Belaidžiai tinklai neretai turi labai ribotą pralaidumą (pvz. 9600 bps) ir didelį vėlinimą (laikas tarp užklausos ir atsakymo)
- Belaidžiai tinklai gali palaikyti arba nepalaikyti IP

Kadangi belaidžiai įrenginiai darosi vis galingesni, tikėtinas variantas jog įmanoma tai, kad WAP ir žiniatinklio modeliai kada nors susilies į bendrą standartą. Taip pat įmanoma ir tai, jog modeliai išsiskirs, nes mažų, belaidžius tinklus naudojančiu įrenginių paskirtis gali išlikti specifinė. Kadangi WAP modelis yra pagrįstas sluoksnine architektūra, jis bus lengvai pritaikomas ateities reikmėms – kad ir kokios jos būtų.

8.2 Elektroninės komercijos analizė

8.2.1 Elektroninės komercijos samprata

Elektroninę komerciją galima būtų apibrėžti kaip verslo formą, kada šalys bendrauja elektroniniu būdu, be fizinio ryšio. Elektroninė komercija - bendra sąvoka, aprėpianti verslo sandorius, valdomus elektroniniu būdu, naudojant telekomunikacijų tinklus.

Elektroninė komercija yra kurianti, vadovaujanti ir plečianti komercinius santykius internetu. Šis naujas verslas pasižymi sparčiai besiplečiančiomis pasiūlos galimybėmis, didėjančia visuotine konkurencija bei milžiniškais vartotojų lūkesčiais. Visame pasaulyje verslas keičia savo organizacines struktūras bei operacines formas: sena hierarchija pamažu nyksta, mažėja barjerų tarp įmonės klientų ir tiekėjų. Kad būtų įveiktos įsisenėjusios kliūtys, verslo procesai yra reorganizuojami, o į pačią reorganizaciją dažnai įtraukiama visa įmonė, jos partneriai, klientai ir net tiekėjai. Elektroninė komercija yra priemonė sudaryti sąlygas tokiems pasikeitimams bei juos paremti pasauliniu mastu. Ji leidžia įmonėms efektyviau ir lanksčiau atlikti vidaus operacijas, artimiau dirbti su tiekėjais bei jautriau reaguoti į klientų poreikius ir lūkesčius.

Sėkmingi elektroninės komercijos sumanymai gali apimti pirkimus, plėtrą ir produktų projektavimą, vadovavimą produkcijai ar gamybos rinkodarą, pardavimus, aptarnavimą, bendradarbiavimą versle, produktų platinimą, mokslinius tyrimus, informacijos skleidimą, komercinių bendruomenių steigimą, mokymą, renginius ir dar daug kitų verslo sferų. Pateiksime keletą elektroninės komercijos veiklos pavyzdžių:

- vartotojai apie produktus daugiau sužino internete prieš pirkdami juos „realiame pasaulyje“;
- vartotojai užsisako produktus tinklu ir juos gauna visuomeniniu pristatymu (paštu) ar per internetą;
- studentai dalyvauja nuotolinio mokymo programose internete (*on-line*) ir taip gauna išsilavinimą ar įgyja profesiją;
- piliečiai, tinklu bendraudami su valstybės institucijomis, pakeičia savo vairuotojo pažymėjimus, registruoja automobilius, moka mokesčius, prašo leidimų statybai ar dalyvauja kituose procesuose;
- firmos parduoda produktus ar paslaugas vartotojams arba kitoms firmoms;
- firmos randa tinkle projektus arba atsisiunčia elektroninius failus (vaizdelius, duomenis, įrašus arba tekstinius failus) per internetą;
- firmos teikia techninę informaciją arba klientus aptarnauja 24 valandas per dieną 7 dienas per savaitę;

- internete pateikiama informacija apie pramogas ir kitus įvykius;
- vyriausybė ir jos institucijos apdoroja prašymus ir pasiūlymus ir kitus internete pateiktus dokumentus.

Vienas iš elektroninės komercijos atvejų būtų elektroninė prekyba. Dabar elektroninė prekyba yra viena iš perspektyviausių ir pažangiausių interneto technologijų. Elektroninę prekybą galima skaidyti į didmeninę, kai tiekiamas didelis prekių ar paslaugų užsakymas, ar mažmeninę, kai klientas dažniausiai yra tiesioginis vartotojas. Vis dėlto, nors šie specialūs atvejai yra didelės ekonominės svarbos, jie yra tik bendro elektroninio verslo operacijų modelio pavyzdžiai. Kiti nemažiau svarbūs pavyzdžiai galėtų būti įmonės vidinės transakcijos arba informacijos keitimasis tarp įmonių.

Daugelis žmonių elektroninę komerciją supranta kaip tradicinį pardavimą internete. Pažvelkime iš kitos pusės. Elektroninės komercijos sumanymas gali būti ne pardavinėti klientams internete, bet aptarnauti ir suteikti informaciją jau dalyvaujantiems elektroninėje komercijoje. Panagrinėkime pavyzdžius:

- įmonė, kuri verčiasi žvyro pristatymu klientams, užsakymus kitai dienai gali priimti savo interneto svetainėje bet kuriuo laiku – dieną ar naktį;
- socialinė aprūpinimo ne pelno siekianti įstaiga rengia paaugliams diskusijas, kur klausimai pateikiami anonimiškai elektroniniu būdu ir atsakymai yra pateikiami interneto svetainėje, kur kiekvienas gali pamatyti atsakymus;
- mažos antikvarinės parduotuvės įdeda į interneto svetainę savo katalogą, kad galėtų parodyti savo naujienas pirkėjams visame pasaulyje;
- bendruomenės organizacija per internetą organizuoja savo narių susitikimus, pateikia informaciją apie bendruomenės paslaugas, įvyksiančius renginius ir kt.

Elektroninės komercijos, kuri neteikia tiesioginės naudos, bet plečia ir įtvirtina jūsų verslą, pavyzdžiai gali būti:

- informacijos apie savo įmonę pateikimas, palengvinant bendradarbiavimą su įmonės pirkėjais, klientais, darbo ieškotojais ir kitais norinčiais bendrauti su įmone;
- pardavimo ciklo sutrumpinimas, pateikiant išsamią informaciją apie produktus. Internete galima pasiūlyti produktus tiems klientams, kurie kitu atveju gali būti nepasiekiami;
- siūlo aukščiausios kokybės klientų aptarnavimą internete;
- pagreitina bendravimą tarp verslo partnerių.

Elektroninė komercija - pasikeitimų technologija. Įmonės, kurios supras ją tik kaip jau egzistuojančių operacijų priedą, elektroninė komercija teiks ribotą naudą. Tuo tarpu įmonėms, norinčios keisti savo organizaciją bei verslo procesus, ji atvers naujų galimybių ir duos daugiausia naudos.

8.2.2 Elektroninės komercijos kategorijos

Elektroninė komercija gali apimti įvairias informacijos judėjimo bei sąveikos sferas. Norint įžvelgti plačias elektroninių priemonių taikymo galimybes, pravartu į elektroninę komerciją pasižiūrėti keliais aspektais. Taigi pagal elektroniniu būdu bendraujančias šalis elektroninę komerciją galima skirstyti į tokias kategorijas:

- Verslas – verslui
- Verslas – vartotojui
- Vartotojas – vartotojui
- Valstybinė institucija – verslui
- Valstybinė institucija – vartotojui
- Valstybinė institucija – valstybinei institucijai

Verslas - verslui kategorija aprėptų įmonių tarpusavio bendravimą elektroninėmis priemonėmis. Pavyzdžiui, įmonė, naudojanti tinklą susisiekti su savo tiekėjais, užsakyti prekes, pasikeisti dokumentais bei atsiskaityti elektroniniu būdu. Visa tai yra pasiekiama 24 valandas per parą 7 dienas per savaitę. Tokia elektroninė komercija pasaulyje sėkmingai vyksta jau keletą metų, ypač vadovaujantis Elektroninės informacijos mainų (*Electronic Data Interchange*) protokolu per privačius ar specialiai tam sukurtus tinklus. Elektroninės priemonės pirmiausia leidžia įmonėms tobulinti savo vidaus operacijas, operatyviau reaguoti į viena kitos poreikius, suaktyvinti bendradarbiavimą, padidinti efektyvumą, sukurti naujų elektroninių verslo paslaugų.

Verslas - vartotojui kategorija daugiausia nusako elektroninę mažmeninę prekybą, todėl dažnai vartotojas, išgirdęs apie elektroninę komerciją, įsivaizduoja būtent šios kategorijos apimtį, nors tai tėra tik viena elektroninės komercijos sričių.

Vartotojas - vartotojui kategorija aprėpia elektroninius vartotojų tarpusavio santykius. Tai gali būti informacijos apsikeitimas tinklu arba elektroniniai aukcionai.

Valstybinė institucija - verslui kategorija nusako elektroninį bendradarbiavimą tarp verslo ir valstybinių institucijų. Pavyzdžiui, viešų valstybės aktų skelbimas internete, kur įmonės savo nuomonę reiškia elektroniniu būdu. Ateityje ši sritis įtrauktų verslo dokumentų tvarkymą, siuntimą bei registravimą tinklu, kasdienių transakcijų, kaip PVM grąžinimas ir daugelio kitų biurokratinių operacijų elektronizavimą. Tai leistų sparčiau bendrauti, mažinti transakcijų išlaidas ir valstybės reguliavimą.

Valstybinė institucija - vartotojui kategorijos pavyzdžių daugiau pasirodys ateityje, kai sparčiai besiplečiančios verslas-vartotojui bei valstybinė institucija-verslui sritys pastūmės valstybę plėtoti savo elektroninę veiklą tokiose srityse, kaip informacijos skleidimas, mokesčių, sveikatos apsaugos ar švietimo programų įgyvendinimas. Jau dabar egzistuojantis pavyzdys – patogus ir sutaupantis daug laiko pajamų deklaravimas elektroniniu būdu.

Valstybinė institucija - valstybinei institucijai sritis aprėps valstybės valdymo bei administravimo perorganizavimą panaudojant informacines technologijas. Jau dabar pasaulyje matyti vadinamosios „Elektroninės vyriausybės“ strategijos užuomazgų, kurios įgyvendinimas lems vyriausybės veiklos kitimą taikant elektroninio verslo metodus valstybiniame sektoriuje. Kitimas įtrauks bendravimą tarp valstybinių institucijų, centrinės ir vietinės valdžios sprendimų priėmimą. Tai turėtų lemti didesnį informacijos valdymo tikslumą bei efektyvumą, mažesnes transakcijų išlaidas, operatyvesnę informacijos kaitą.

8.2.3 Elektroninės komercijos įtaka

Elektroninė komercija nėra vien ateities vizija. Tai vyksta šiandien. Pasauliniu mastu elektroniniai procesai itin spartėja. Jau yra daug elektroninio verslo sėkmės pavyzdžių, tarp lyderių - JAV, Japonija bei Europos šalys. Tuo tarpu elektroninės informacijos mainų susitarimai, sparti interneto bei technologijų plėtra daro didelę įtaką pasaulio raidai.

Elektroninė komercija turės įtakos tiek verslui, tiek visai visuomenei. Puikiai panaudojančioms savo potencialą įmonėms, elektroninė komercija atvers visiškai naujų galimybių, kai radikalūs pasikeitimai turės įtakos vartotojų lūkesčiams, pakeis jų požiūrį į rinką ar net sukurs naujų rinkų. Visi rinkos ir vartotojų pokyčiai paveiks verslo dalyvius, net ir tuos, kurie ignoruos informacinių technologijų įtaką.

Kita vertus, individualūs asmenys atras naujų būdų pirkti, gauti informaciją ar pasinaudoti paslaugomis, bendrauti su visuomene, nepaisant geografinių ar laiko ribų. Elektroninės komercijos populiarumas ateityje būtų lygintinas su automobilių populiarumu ar telefono ryšio paplitimu.

8.2.4 Elektroninės komercijos veikla

Elektroninės komercijos sprendimai taikomi įvairioms verslo sferoms:

- prekybai, pardavimams ir reklamai;
- tiekimui;
- finansavimui ir draudimui;
- prekybiniams sandoriams: užsakymams, pristatymams, mokėjimams;
- produkto pateikimui;
- produkto vystymui;
- dviejų organizacijų bendradarbiavimui;
- valstybės reguliavimui (lengvatos, leidimai, mokesčiai, muitai ir kt.);
- logistikai;
- skaitmeniniam prekių pardavimui;
- apskaitai;

- ginčų sprendimui.

Visas komercinio sandorio, įskaitant užsakymą, transportavimą, dokumentų tvarkymą ir mokėjimą, ciklas gali būti atliekamas elektroniniu būdu. Vis dėlto elektroninėje komercijoje reikia atsižvelgti į daugelį tokių aspektų, kaip informacijos saugumas, asmens teisių apsauga, teisiniai klausimai ir procedūros.

Kalbant apie elektroninę prekybą, reikėtų pabrėžti skirtumą tarp elektroninės prekybos materialiomis prekėmis ir paslaugomis bei elektroninės prekybos elektroninio pavidalo informacija, kai prekė arba paslauga gali būti siunčiama tiesiogiai tinklais (vaizdai, balsas, tekstas, programinė įranga ir pan.).

Elektroninė materialių prekių ir paslaugų prekyba atspindi dabartinių pardavimo būdų evoliuciją, kai naujos technologijos leidžia didinti verslo produktyvumą bei efektyvumą, ir dėl to mažėja išlaidos, plečiasi rinkos galimybės, geriau aptarnaujami vartotojai, skatinamos naujovės. Ši elektroninės prekybos forma labai didins konkurenciją versle, bet neturėtų smarkiai paveikti šalies darbo lygio.

Elektroninio produkto pardavimas (programinė įranga, muzika, vaizdo įrašai, daugialypės darbai, žaidimai ir kt.) atspindi kokybiškai naują prekybos būdą, kai visas prekybos sandorio ciklas gali būti atliekamas per internetą.

Elektroninės komercijos nauda verslui:

- mažesnės reklamos išlaidos;
- mažesnės tiekimo kainos, ypač prekėms, kurios gali būti tiekiamos elektroniniu būdu;
- sumažintos projektavimo ir gamybos kainos;
- geresnės rinkos pažinimas ir strateginis planavimas;
- daugiau rinkos galimybių;
- geresnis priėjimas prie rinkos;
- priėjimas prie naujos rinkos;
- klientų informavimas apie produktų ir paslaugų naujoves.

Rinkos tyrimai, naujų galimybių analizė, informacija teisės klausimais, automatinis rinkos duomenų generavimas - viskas gali būti atliekama elektroniniu būdu.

Bendradarbiavimą tarp įmonių būtų galima palengvinti bendraujant verslo kanalais, taip gerinamas valstybinės ir vietinės informacijos perdavimas. Tarp įmonių ir klientų gali būti įvairiais būdais bendraujama, pvz., tinkle esama reklama ar per elektronines parduotuves, kuriose kompanijos gali pateikti išsamią informaciją apie jų produktus ir paslaugas, taip pat ir techninę informaciją, naudojimosi taisykles ir atsakymus į rūpimus klausimus. Visa tai galima pateikti lengvai suprantamais ieškojimo būdais.

Paskutiniais metais, stengiantis pagerinti verslo efektyvumą, vis geresnių rezultatų pasiekė bendravimas tarp įmonių ir klientų. Verslo procesai plinta už įmonės ribų, ir kiekviena įmonė turi savo

proceso dalį, pavyzdžiui, virtuali įmonė, kur kiekviena dalyvaujanti įmonė žaidžia pagal savo taisykles, bet tuo tarpu draugiškai bendrauja įmonių tinku, atsižvelgdama į rinkos galimybes.

Ten, kur įmonės gali kartu sukurti vieną virtualią įmonę, apimančią viską, t. y. nuo prekių gamybos ir paslaugų plėtojimo iki pardavimų, tikimasi, kad daug pasikeitimų atsitiks susietoje industrijos struktūroje. Vienas iš pavyzdžių yra televizijos parduotuvės, kuriose bandoma išvengti mažmeninės prekybos paslaugų tarp vartotojų ir prekių gamintojų. Todėl ryšiai tarp gamintojų ir mažmenininkų sektorių tampa vis mažiau svarbūs ir kt.

Pagrindiniai verslo strategijos pavyzdžiai, pagrįsti elektronine komercija:

- elektroninės pardavimo vietos buvimas: reklama, bendravimas TV / Tinklinės (*on-line*) parduotuvės;
- efektyvus vartotojų atsiliepimų tvarkymas;
- prekyba elektroniniu būdu;
- tiekimo grandinės valdymas.

Potencialiai elektroninė komercija gali visiškai sutvarkyti tiekimą verslo proceso dalyviams, nepaisant dalyvių geografinės padėties ir laiko zonos.

8.2.5 Tiekėjų galimybės ir klientų nauda

Kaip matyti iš lentelės, elektroninė komercija suteikia keletą galimybių tiekėjams ir tam tikrą naudą klientams.

2.1 lentelė Elektroninės komercijos teikiamos galimybės bei nauda

Tiekėjo galimybės	Vartotojo nauda
Pasaulinė prieiga	Pasaulinis pasirinkimas
Didesnė konkurencija	Aptarnavimo kokybė
Vartotojų skaičius	Reikalinga prekė ir paslauga
Greičiausia tiekimo grandinė	Greitas atsakymas į pageidavimus
Išlaidų mažinimas	Mažesnės kainos
Naujos verslo galimybės	Nauji produktai ir paslaugos

Pasaulinę prieigą / pasaulinį pasirinkimą

Elektroninės komercijos ribos nėra nustatytos pagal geografinę ar valstybinę padėtį, bet greičiau pagal kompiuterių tinklų išplitimą. Elektroninė komercija suteikia galimybę ne tik smulkiems tiekėjams būti pasauliniame tinkle, bet ir verstis verslu pasauliniu mastu.

Vartotojų naudos požiūriu, tai yra pasaulinis pasirinkimas - vartotojas gali išsirinkti iš visų potencialių tiekėjų reikalingiausią prekę arba paslaugą, nepaisydamas geografinės padėties.

Didesnis konkurencingumas / aptarnavimo kokybė

Elektroninė komercija suteikia tiekėjams galimybę efektyviau konkuruoti bendraujant su klientais. Įmonės, dirbančios elektroniniu būdu, gali pasiūlyti klientams geresnį pardavimo aptarnavimą, suteikti daugiau informacijos apie produktą ir greitai atsakyti į visus rūpimus kliento klausimus. Todėl klientai geriau ir greičiau aptarnaujami.

Vartotojų skaičius/ Reikalinga prekė ir paslauga

Elektroniniu būdu tiekėjai gali greitai surinkti detalią informaciją apie kiekvieno kliento norus ir juos įvykdyti. Be to, klientui reikalingas produktas tiekėjų bus pasiūlytas rinkos kaina. Pavyzdžiui, žinomas atvejis, kaip tinkle esantis vienas žurnalas kiekvienam skaitytojui yra padaręs priėjimą prie jų dominančių straipsnių, o jo perskaitytus straipsnius ištrina.

Greičiausia tiekimo grandinė/greitas atsakymas į pageidavimus

Elektroninė komercija leidžia sutrumpinti tiekimo grandinę. Keletas pavyzdžių: prekės iš gamyklos yra siunčiamos tiesiai vartotojui, t. y. nereikia perpardavinėtojų (mažmenininkų) paslaugų ir jų antkainio. (Paprasčiausiai nėra elektroninės komercijos mokesčio, bet ieškant produkto kitu būdu, t. y. laikraščiuose, kataloguose, telefonais ir kt. būdais, tai kainuoja ir užtrunkama daug laiko).

Tinkamiausias pavyzdys būtų tų produktų ir paslaugų, kuriuos galima tiesiogiai elektroniniu būdu atsiųsti, vaizdo įrašai, muzika, laikraščiai, žurnalai, programinė įrangą, įvairi informacija ir kt. Taigi elektroniniu būdu gaunant prekes ir paslaugas tiekimo grandinė gali būti visiškai panaikinama.

Atitinkamai ir vartotojui yra iš to nauda, nes galima greitai įsigyti norimą produktą, apeinant bendravimą su kitomis įmonėmis ir vietiniais tiekėjais.

Išlaidų mažinimas / geresnės kainos

Bet koks verslas, kuris įtraukia bendravimą su žmonėmis, reikalingas tam tikrų išlaidų, tačiau bendraujant elektroniniu būdu galima sumažinti išlaidas ir dėl to pasiūlyti vartotojams mažesnes produktų ir paslaugų kainas.

Naujos verslo galimybės / Nauji produktai ir paslaugos

Elektroninė komercija apima ne tik esamų prekių ir paslaugų rinką, bet ir suteikia galimybę visiškai naujiems produktams ir naujoms paslaugoms įeiti į rinką. Pavyzdžiui, naudojimasis tinklu ir aptarnavimo paslaugos, direktorijos, bendradarbiavimus elektroniniu būdu ir daugelis kitų informacijos teikimo paslaugų.

Daugelis galimybių ir reikalavimų yra skirtingi, bet iš dalies ir tarpusavyje susiję. Pavyzdžiui, didėjanti konkurencija pagerins aptarnavimo kokybę, o sutrumpinta tiekimo grandinė sumažins papildomas išlaidas ir sumažins kainas.

8.2.6 Elektroninės komercijos pavyzdžiai Lietuvoje

Čia pateikta keletas elektroninės komercijos pavyzdžių Lietuvoje. Kadangi viešai lengviausia prieiti prie elektroninių parduotuvių – jos sudarys didžiąją dalį pavyzdžių.

Mažmeninės prekybos pavyzdžiai (el. parduotuvės):

- www.super.lt - interneto parduotuvė Super.lt yra uždarosios akcinės bendrovės „Naujosios komunikacijos prekyba“ projektas. Čia itin platus knygų pasirinkimas, kainos mažesnės nei knygynuose, o atsiskaitoma šiais būdais: grynais pinigais, banko pervedimu, čekiu, per „Hansabanką“ internete „hansa.net“, per Vilniaus banką internete „VB Internet@s“.
- www.venera.lt – intymių prekių interneto parduotuvė. 1-osios vietos laimėtojas 8-ajame Lietuvos svetainių čempionate. Nominacija „E-prekyba, verslas, komercija“.
- http://www.eks.lt – kompiuterių dalių interneto parduotuvė, atrodo jog ne visai užbaigta, nors viskas ir veikia; UAB „Ineksmė“.
- http://www.prestige.lt - čia galite apžiūrėti, išsirinkti iš daugybės rūšių papuošalų; UAB „Dzūkijos perlas“.
- http://www.diabetas.lt - diabeto prekės internetu - platus asortimentas, pigu, patogu užsakyti, pristatymas per 2 darbo dienas visoje Lietuvoje.
- http://www.naminukas.lt – interneto parduotuvė, aukcionas, sendaikčių turgus, labai gera varotojo sąsaja; UAB „Pretendentas“.
- http://www.pirkite.lt – neužbaigta elektroninė parduotuvė (nors viskas veikia), šiuo metu parduodama.
- http://www.commerce.lt – šią parduotuvę galima laikyti pavyzdžiu kaip nereikia kurti elektroninių parduotuvių – nepatogi, pritaikyta tik Internet Explorer naršyklei. UAB „Penki kontinentai“.
- http://www.books.lt – nors naudojama ta pati sistema kaip ir www.commerce.lt („*Trade manager*“ v1.0) čia viskas veikia korektiškai.

Elektroninės paslaugos:

- http://www.fotofabrikas.lt - Fotografijos paslaugų portalas:
  - Nuotraukų ikėlimas
  - Portale galite turėti 30MB talpos nemokamą albumą.
  - Užsiregistravę ir įkėlę savo nuotraukas prie savo albumo galėsite prisijungti bet kuriuo paros metu, iš bet kurios pasaulio vietos – įsikelti, siųsti, paviešinti ar kitaip valdyti norimas fotografijas.
  - Nuotraukų gamyba
  - Nuotraukų gamybos užsakymas internetu labai paprastas, greitas ir patogus.
  - Užsiregistravę vartotojai savo sukeltas nuotraukas gali surasti „Mano Fabrikas“ skiltyje - tereikia pažymėti, kurias norite gaminti, pasirinkti pageidaujamus parametrus - tą galite padaryti neatsikėlę nuo kėdės.

    - Dovanų ir suvenyrų gamyba (su nuotraukomis)
    - El. prekyba foto prekėmis
    - Reklamos paslaugos
- http://www.foto.lt – pasak savininkų – „interneto portalas, skaitmeninių technologijų dėka išplečiantis tradicinę fotografijos sąvoką ir teikiantis moderniausias šios srities paslaugas“. Tačiau šis portalas ne toks patogus kaip http://www.fotofabrikas.lt, taip pat siūlo mažiau paslaugų. Naujoji portalo versija leidžia svetainę naršyti visomis naršyklėmis, tačiau iki 2004.05.01 tai buvo galima daryti tik su Microsoft Internet Explorer naršykle. Pabandžius svetainę atidaryti su kita naršykle, būdavo parašyta kad būtina būtent Microsoft Internet Explorer naršyklė. Tačiau „apėjus“ apsaugą svetainė sėkmingai veikdavo ir su Mozilla Firefox, tai rodo iš principo neteisingą požiūrį į portalų kūrimą.
- http://www.cv-online.lt – kokybiškiausias įdarbinimo agentūros portalas Lietuvoje. Patogus tiek darbdaviui tiek vartotojui, aiški ir paprasta struktūra, daugiausia vartotojų (per 66.000 CV, daugiau kaip 2.5mln lankytojų per mėnesį). Ne tokie vykę bandymai - http://www.cv.lt bei http://www.cvmarket.lt

Elektroninė bankininkystė:

- http://lt.hanza.net – AB „Hansabankas“ elektroninio banko sistema – gerai realizuota, patogi ir greita sistema. Per metus kasdieninio naudojimo neteko susidurti su jokiais keblumais ar sutrikimais. Čia galima ne tik tvarkyti banko sąskaitas bei pervedinėti ar konvertuoti pinigus, tačiau ir apmokėti komunalinius mokesčius, gauti pažymą pajamų deklavarimui apie sumokėtas palūkanas arba turimas lėšas „Hansabanke“ bei gauti daug kitų paslaugų.
- https://ebankas.vb.lt – visiška priešingybė AB „Hansabankas“ sistemai – VB Internet@s. Ši sistema nepatogi, lėtai veikia (o neretai iš viso neveikia), informacija vėluoja, dažnai apie pervestus pinigus informacijos nebūna, nors operacija ir įvykdyta. Kartais galima pamatyti sistemines klaidas, kurių išvedimas – ne sistemos autorių darbas.

Kiti elektroninės komercijos pavyzdžiai:

- http://deklaravimas.vmi.lt – Elektroninio deklaravimo sistema (Gyventojų pajamų deklaravimo modulis) – labai kokybiškai ir patogiai vartotojui padaryta sistema. Kadangi ši sistema Lietuvoje yra naujiena, panagrinėsime ją plačiau:
    - Naudojimosi EDS privalumai:
        - Galite pildyti ir teikti deklaracijas bet kuriuo paros metu;
        - Jums nereikia vykti į AVMI teritorinį skyrių ir teikti popierinio formato deklaraciją;

- Jei esate interneto bankininkystės vartotojas(a), tuomet nereikia vykti į VMI netgi vartotojo registracijai, Sutartį galite sudaryti elektroniniu būdu;
- Gaunate patogias deklaracijų pildymo ir tikrinimo priemones;
- Operatyviai gausite elektroninį pranešimą apie deklaracijos priėmimą, arba padarytas klaidas.

- Kaip tai vyksta?
  - Jungiatės prie EDS interneto svetainės iš savo namų arba darbovietės;
  - Atsisiunčiate deklaracijų formų šablonus ir jų pildymo priemones;
  - Registruojatės pastoviu EDS vartotoju;
  - Pildote deklaracijas savo kompiuteryje ir, užbaigę pildymą, pateikiate jas į EDS internetu;
  - Informaciją apie deklaracijos priėmimo rezultatą gaunate elektroniniu paštu;
  - Pateiktų deklaracijų istoriją galite peržiūrėti prisijungę prie sistemos savo vardu.

8.2.7 Elektroninės komercijos sprendimai

Elektroninės komercijos sprendimai gali būti įgyvendinti įvairiais lygiais - nuo paprasčiausio įmonės įjungimo į elektroninį tinklą iki sudėtingų elektroninių verslo procesų palaikymo.

Elektroninė komercija padeda Lietuvoje atrasti naujas rinkas, naujas galimybes mažoms ir didelėms įmonėms pasauliniu mastu. Elektroninės komercijos strategijos tikslai yra padėti Lietuvos kompanijoms sukurti stipresnius ryšius su vartotojais bei verslo partneriais, nes geri santykiai su vartotojais bei pardavimo partneriais yra verslo sėkmė.

Elektroninės komercijos pranašumai yra mažesnės kainos, geriau pasiekiami vartotojai, spartesni atsakymai, didesnis pasirinkimas. Internetas yra geras susisiekimo kanalas - jis yra greitas, patikimas, nebrangus ir plačiai prieinamas, taip stiprinantis santykius su klientais bei partneriais.

8.3 Elektroninės parduotuvės veiklos analizė

8.3.1 Įvadas

Dažniausiai sutinkama elektroninės komercijos apraiška pasauliniame interneto tinkle – elektroninės parduotuvės. Tačiau ši rinka jau perpildyta, o naujovių siūloma vis mažiau.

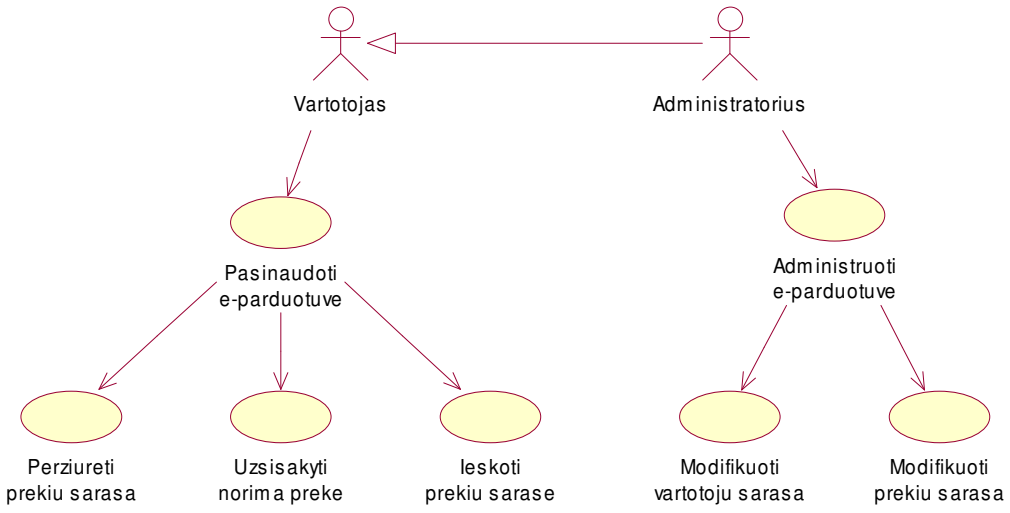
Šiuo metu itin sparčiai vystosi delninių kompiuterių paklausa – jų galimybės nuolat didėja, o kainos mažėja, todėl vis daugiau vartotojų iškeičia nešiojamąjį kompiuterį į delninį. Dauguma šių kompiuterių turi GPRS arba HSCSD duomenų perdavimo GSM tinklais galimybę, o kai kurie – net WLAN palaikymą. Šiais delniniais kompiuteriais galima prisijungti prie pasaulinio internet tinklo, ir naudotis jo teikiamomis galimybėmis. Dažnai tai daroma WAP protokolu, naudojant WML kalbą. Šias technologijas palaiko ir dauguma naujų mobiliųjų telefonų.

Todėl šiame darbe bus bandoma sukurti elektroninės parduotuvės prieigą delniniams kompiuteriams, bei jei tai bus įmanoma padaryti naudojant WAP/WML technologijas – ir mobiliesiems telefonams.

8.3.2 Elektroninės parduotuvės panaudojimo atvejų modelis

Šiame skyriuje pateikiami pradiniai funkciniai reikalavimai elektroninei parduotuvei, jie gali keistis darbo metu. Išsamiau žr. skyrių „Projektavimas - Bendrieji sistemos reikalavimai“.

Bendrasis panaudojimo atvejų modelis



Bendrasis panaudojimo atvejų modelis

Pagrindiniai aktoriai:

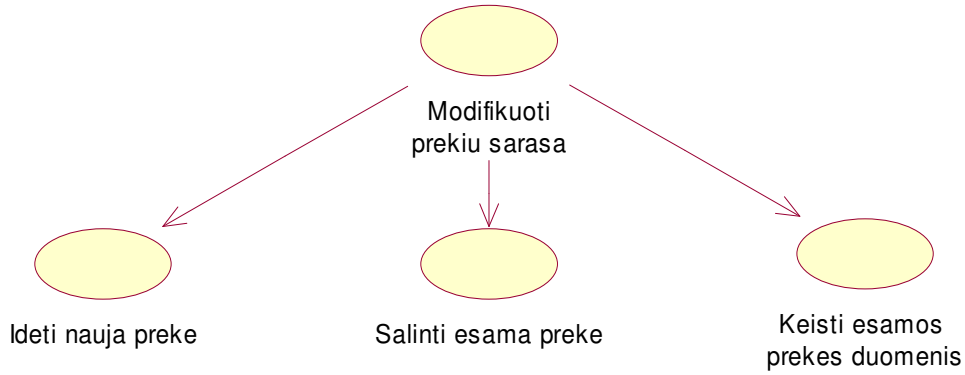
- Vartotojas – informacinėje sistemoje prekės užsakymo metu registruojamas asmuo, galintis peržiūrėti prekių sąrašą, atlikti jame paiešką, bei užsakyt prekes.

- Administratorius – tai informacinėje sistemoje užregistruotas asmuo, galintis modifikuoti prekių bei vartotojų sąrašus, t.y. keisti esamus, juos šalinti arba sukurti naujus.

Pagrindiniai panaudojimo atvejai:

- Pasinaudoti el. parduotuve
  - Peržiūrėti prekių sąrašą
  - Užsisakyti norimą prekę
  - Ieškoti prekių sąraše
- Administruoti el. parduotuvę
  - Modifikuoti vartotojų sąrašą (gali atlikti tik administratorius)
  - Modifikuoti prekių sąrašą (gali atlikti tik administratorius)

Prekių sąrašo modifikavimo modelis

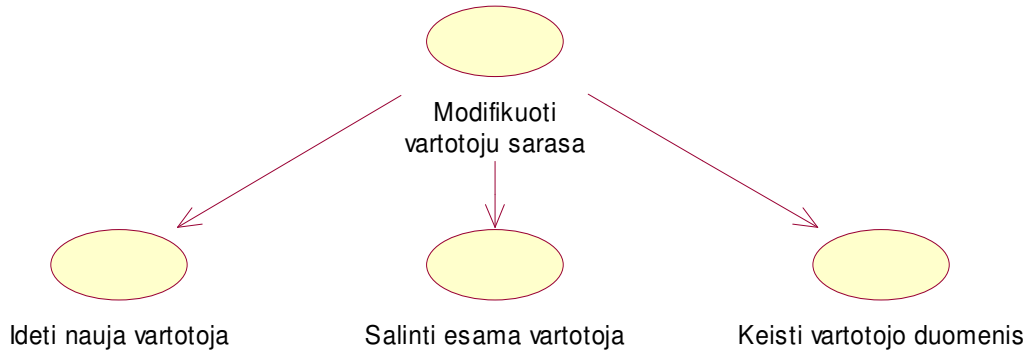


Prekių sąrašo modifikavimo modelis

Prekių sąrašo modifikavimo panaudojimo atvejai:

- Įdėti naują prekę (gali atlikti tik administratorius)
- Šalinti esamą prekę (gali atlikti tik administratorius)
- Keisti esamos prekes duomenis (gali atlikti tik administratorius)

Vartotojų sąrašo modifikavimo modelis



Vartotojų sąrašo modifikavimo modelis

Vartotojų sąrašo modifikavimo panaudojimo atvejai:

- Įdėti naują vartotoją (gali atlikti tik administratorius)
- Šalinti esamą vartotoją (gali atlikti tik administratorius)
- Keisti esamo vartotojo duomenis (gali atlikti tik administratorius)

8.3.3 Veiklos analizės išvados

Pagal elektroninės parduotuvės panaudojimo atvejų modelį sudarytą ankstesniame skyriuje, galima daryti išvadą jog sudarėme funkcinius reikalavimus atitinkančius standartinės elektroninės parduotuvės stereotipą. Kuriant panaudojimo atvejus, ir tuo pačiu funkcinius reikalavimus buvo atsižvelgta į tai jog elektroninė parduotuvė turės veikti ir WAP/WML terpėje, bei atitikti šių technologijų apribojimus, todėl buvo taikomas nesudėtingas modelis. Kiekvieną aprašytą panaudojimo atvejį bus įmanoma realizuoti abiejose darbo terpėse – tiek HTTP/HTML, tiek WAP/WML.

Taip pat pagal elektroninės parduotuvės panaudojimo atvejų modelį sudarytą ankstesniame skyriuje, galima daryti išvadą jog darbo tema pasirinkta teisingai, bei darbo atlikimas panaudojant WAP/WML technologijas kaip vieną iš nešančiųjų terpių yra įmanomas, o tuo pačiu ir elektroninės parduotuvės pasiekimas bei naudojimas iš daugumos mobiliųjų įrenginių, įskaitant mobiliuosius telefonus.