

Network Service Maps

Vorstellung
Diplomarbeit

Background

- ▶ Increasingly heterogeneous network environments
 - WLAN, 3G, 2.5G, WiMAX
 - numerous provider
 - community-based WLAN hotspots not “always-on”
 - mobile devices' number of network interfaces increase
- ▶ Mobile users need to be supported in making optimal network access decision
 - existing solutions for network access finding insufficient
 - need for an efficient, flexible network information service

Service Maps

- ▶ Service maps for describing network services
- ▶ XML-based data format for describing arbitrary services
 - *locations, services, provider* are grouped to *service instances*
 - informal *tags* on each data type for searching
 - *refinements* provide extensibility and flexibility for service-specific information and variation in distribution size

Service Maps: Example

```
<service-map (...)>
```

```
<location id="mzh5-wlan-tcom-1">  
  <tag>Bremen</tag>  
  <tag>Uni-Bremen</tag>  
  <geo-field>  
    <center>  
      <latitude>53.106652</latitude>  
      <longitude>8.852375</longitude>  
    </center>  
    <radius>12</radius>  
  </geo-field>  
</location>
```

```
<provider id="t-com" name="T-Com">  
  <refinement>  
    <ico:icon type="raster" file-type="png">  
      <ico:icon-uri>tcom.png</ico:icon-uri>  
    </ico:icon>  
  </refinement>  
</provider>
```

```
<service id="wireless-tcom">  
  <tag>network-access</tag>  
  <refinement>  
    <charges:charges>  
      <charges:charge type="time"  
        charge-unit="1min"  
        cost-per-unit="0.05"  
        cost-unit="EUR"/>  
    </charges:charges>  
  </refinement>  
</service>
```

```
<instance>  
  <location-reference  
    ref="mzh5-wlan-tcom-1"/>  
  <service-reference ref="wireless-tcom"/>  
  <provider-reference ref="t-com"/>  
</instance>
```

```
</service-map>
```

Internet Media Guides (IMG)

- ▶ IETF distribution platform for meta data providing
 - push transport:
 - IMG Announce
 - IMG Subscribe/Notify
 - pull transport: IMG Query/Resolve
 - container: IMG Envelope with version etc.
- ▶ Concept of IMG Sender, Receiver, Transceiver
- ▶ Suitable platform for distribution of service maps

Pamina Distribution Platform

- ▶ TZI implementation of IMG distribution platform
 - IMG Announce through FLUTE/Muppet
 - IMG Subscribe/Notify through XMPP
 - IMG Query/Resolve through HTTP-Get
 - Upload through HTTP-POST
- ▶ Modular platform, combining of modules to IMG sender or IMG receiver
- ▶ Only simplistic store-and-forward IMG transceiver yet

Missing Parts for Deployment

- ▶ With the mentioned components existing, let's create what's missing for serious deployment:
 - Business cases
 - More sophisticated infrastructure component
 - In-infrastructure announcer
 - Authentication for service maps
 - Service map URN and its resolution
 - Network service maps client GUI

Business Cases

- ▶ Community-based network information service
 - offer flexible and inexpensive service
 - give more reliable, up-to-date data
 - trust model for user-submitted data, prevent spam
 - e.g. *slashdot*-like centralised user “karma” based on positive or negative feedback on contribution
- ▶ Commercial provider-based business
 - convince network providers to employ service maps
 - higher customer loyalty
 - save bandwidth
- ▶ Develop adequate service maps architecture

Infrastructure Component

- ▶ More sophisticated infrastructure component than store-and-forward for benefiting from service maps
- ▶ Caching
 - acquire and redistribute non-local service maps requested by clients
 - automatically keep track of changes
- ▶ Filtering and aggregation
 - suitable filter syntax and algorithms (indexing in DB?)
 - scheduling of broadcast according to statical configuration and client requests
 - filter expression for IMG Query and IMG Subscribe

In-Infrastructure Announcer

- ▶ Access points for WLAN hotspots deployed
- ▶ Service maps announcement from AP saves bandwidth and limits broadcast to local data
- ▶ but: FLUTE sender CPU requirements (FEC etc.)

- ▶ Alternative: light-weight FLUTE relay on AP
 - FLUTE data stream prepared on higher performance server
 - transferred once to AP
 - continuously sent by AP
- ▶ To be implemented for Linksys Access Point

Service Map Authenticity

- ▶ Clients and transceivers need to ensure authenticity
- ▶ Validity after filtering and aggregation in transceivers
- ▶ Canonical form of service map parts and identifiers
- ▶ Usage of PKI, XML Signatures, *hash trees* and *authenticated data structures* – basic procedure:
 - author provides cryptographic hashes for the atomic parts and digitally signs those
 - transceiver filters data, leaves security information as-is
 - receiver verifies authenticity by checking hashes of left atomic parts and digital signature of all hashes (including filtered-out atomic parts)

Service Map URN

- ▶ Basing on IMG URN
 - e.g. urn:img:tzi.org:20061028:campus-wlan
- ▶ Goal: keep service maps' independence from IMG
- ▶ Additional features of service maps URN
 - filter syntax: tags, geo filter, refinement types, (possibly limited) XPath expressions
 - versions, fragments (refinements etc.)
 - canonical form of URN
- ▶ urn:servicemap:tzi.org:20061028:campus-wlan?
?tags=mzh,e5;coord=53.10663,8.852487

Service Map URN Resolution

▶ DDDS with DNS as database

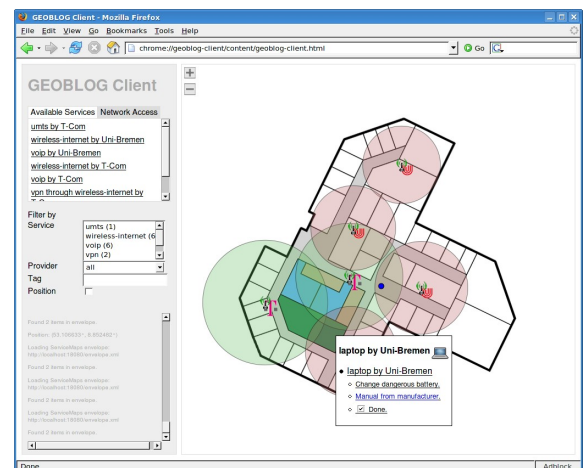
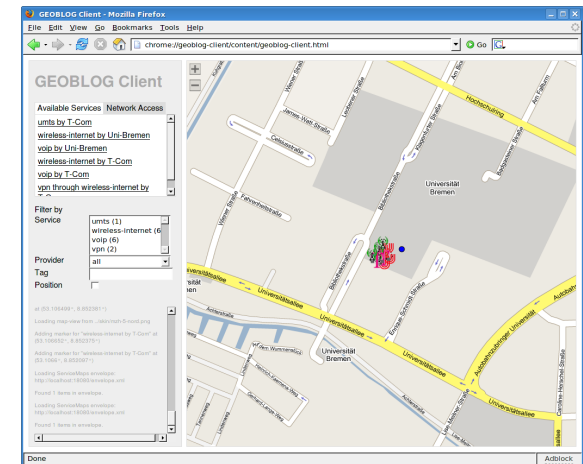
- domain's authoritative DNS offers rewriting rules for supported distribution schemes and protocols

▶ Basic procedure

- start: urn:servicemap:tzi.org:20061028:campus-wlan
- NAPTR lookup for tzi.org might reveal:
tzi.org.
;; ord prf flg srv regexp
IN NAPTR 100 50 "u" "XMPP" "!^(.*)\$!jid:img@dataman/\1!" .
IN NAPTR 100 50 "u" "HTTP" "!^(.*)\$!http://dataman/files/\1!" .
- choosing IMG Subscribe through XMPP leads to URI
jid:img@dataman/urn:servicemap:tzi.org:20061028:campus-wlan
- alternatively, NAPTR could provide resolution service

Network Service Maps Client GUI

- ▶ Display of available services
 - geo-visualisation or
 - currently available services list
- ▶ Navigation software integration
- ▶ Basing on Pamina receiver
- ▶ HTTP to web browser as GUI
- ▶ *wack* integration
 - for finding further hotspots (possibly uploading to community-service)
 - for connecting to AP



Network Service Maps Client GUI (2)

The screenshot displays the GEOBLOG Client interface within a Mozilla Firefox browser window. The browser's address bar shows the URL `chrome://geoblog-client/content/geoblog-client.html`. The main interface is divided into a left sidebar and a central map area.

Left Sidebar:

- GEOBLOG Client**
- Available Services** / **Network Access**
- umts by T-Com
- wireless-internet by Uni-Bremen
- voip by Uni-Bremen
- wireless-internet by T-Com
- voip by T-Com
- vpn through wireless-internet by T-Com

Filter by Service:

- umts (1)
- wireless-Internet (6)
- voip (6)
- vpn (2)

Provider: all

Tag:

Position:

Found 2 items in envelope.

Position: (53.106633°, 8.852482°)

Loading ServiceMaps envelope: `http://localhost:18080/envelope.xml`

Found 2 items in envelope.

Loading ServiceMaps envelope: `http://localhost:18080/envelope.xml`

Found 2 items in envelope.

Loading ServiceMaps envelope: `http://localhost:18080/envelope.xml`

Found 2 items in envelope.

Loading ServiceMaps envelope: `http://localhost:18080/envelope.xml`

Found 2 items in envelope.

Map Area:

The map shows a building complex with several overlapping circular service coverage areas. A popup window is visible over the map, titled **laptop by Uni-Bremen**. The popup contains the following information:

- laptop by Uni-Bremen** (with a laptop icon)
- **laptop by Uni-Bremen**
- ◊ [Change dangerous battery.](#)
- ◊ [Manual from manufacturer.](#)
- ◊ [Done.](#)

The status bar at the bottom of the browser window shows "Done" on the left and "Adblock" on the right.

Summary

▶ Conceptual challenges

- data management: caching, aggregation, filtering; scheduling of broadcasts
- service map authentication
- service map URN: filter expression and resolution
- community-based service maps
- client GUI design

▶ Implementation

- Pamina-based infrastructure component
- in-infrastructure FLUTE relay
- client GUI