



# The Mobile Phone Wizard System

An overview of a distributed  
device and customer  
repository

# Guiding Principles

- Detecting customers, not devices
  - Real-time header analysis and profiling crucial for added value
  - Only way to reliably detect non-mobile browsers and modified handsets
  - Added opportunity to serve premium content to upgraded devices
  - Believe the device, up to a point!

# Guiding Principles

- Make Wizard as simple as possible
  - Minimal client-side coding required
  - “Set it and forget it”
  - Multiple interfaces and documented API (SOAP/POST/Client Software)
  - Simplicity and extensibility by design

# Guiding Principles

- Make Wizard as reliable as possible
  - Multiple distributed servers
  - Modular, so partial nodes can be created.
  - Multiple caching levels for minimal throughput time

# Guiding Principles

- Make Wizard as accurate as possible
  - Data source weighting
  - Data contention
  - Proactive data gathering and correction
  - Intelligent system workload assignment
  - Make the system do the work

# Guiding Principles

- Make Wizard as clever as possible
  - Never calculate the same profile twice globally
  - Automate everything that can be automated reliably and validly
  - Add in real-world information about ISPs, standards, file-types, browsers, JVMs, etc.

## Problems to solve

- UAProfiles are often incomplete or inaccurate
  - Manufacturer inconsistency
  - Deficiencies in standard
  - Vendor-specific extensions
  - Typos
  - UAProfile version control weak

## Problems to solve

- Devices with generic ID string and no UAProf
  - Mozilla-based browsers
  - Some Smartphones

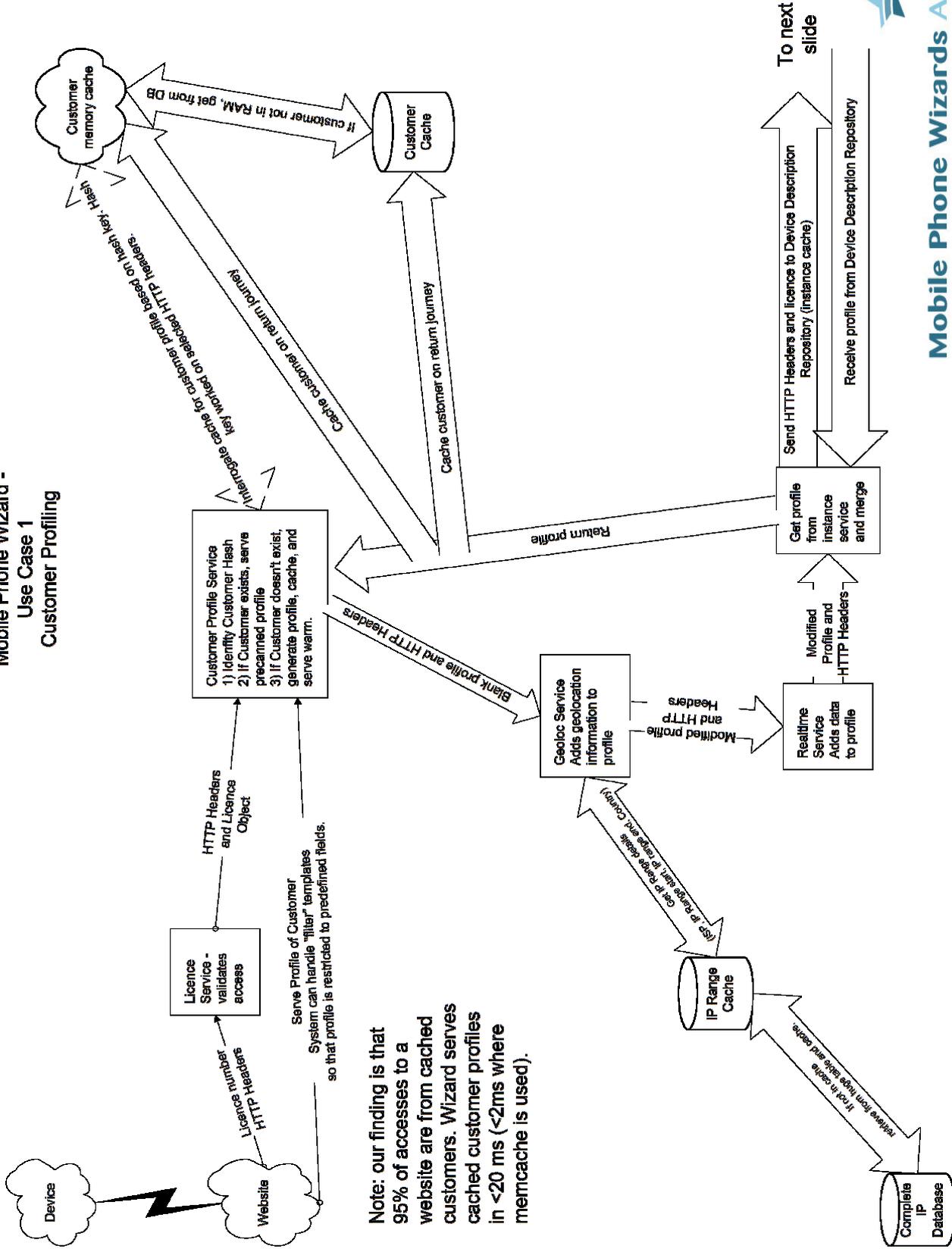
## Problems to solve

- Devices can be upgraded by the user or crippled by the telco.
  - “Ideal” device profile not adequate
  - Real-time header and ISP analysis pretty much mandatory

# Problems to solve

- Creating a profile
  - UAProfile overridden by
  - Manual data overridden by
  - Browser capability data overridden by
  - Real-time analysis

# Mobile Phone Wizard - Use Case 1 Customer Profiling



Mobile Phone Wizards  
 Device Description  
 Repository - profile  
 retrieval from HTTP  
 Headers

Profile field priorities  
 Each field in the profile has potentially many sources. Here is the priority list of information, from least to most important.

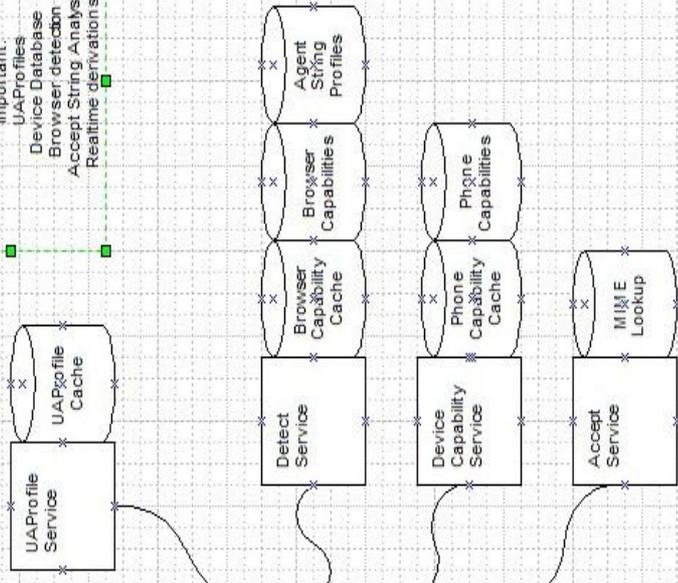
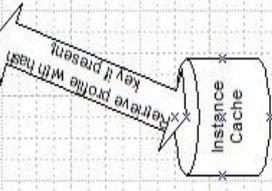
- UAP Profiles
- Device Database
- Browser detection
- Accept String Analysis
- Realtime derivations

Profile Service - generate profile from HTTP Headers.

- 1) Retrieve UAP Profile if it exists, and map to Wizard schema. Add to blank profile.
- 2) Send HTTP Headers to Detect Service to get identifying PHONE ID String (e.g. Nokia2100/1.0). This service also returns a browser profile.
- 3) If the phone model ID has been identified (i.e. if we have manual data for this phone), get cached profile from device service.
- 4) Pass Accept string to the ACCEPT service and superimpose data onto the current profile.
- 5) Add any more, realtime heuristic assumptions about phone.
- 6) Return profile.

Instance Profile Service

- 1) Work out hash based on HTTP\_USER\_AGENT and HTTP\_ACCEPT.
- 2) Retrieve from cache if possible
- 3) If not cached, send HTTP Headers to Profile service.
- 4) If profile is new, cache it



Note: Wizard makes extensive use of cached objects: in our system, they are serialised PHP arrays to avoid processing overhead between PHP and XML.

