Mobile Phone Firmware Update "On-the-Fly"

Bill Kaminsky
University of San Francisco

Background

- Telecom industry in a “slump” including wireless
- Need to spur new business with exciting new services
- 3G is launching
- 2.5G and 3G technologies provide better data capabilities
- Industry is trying to find ways to increase ARPU with new data services

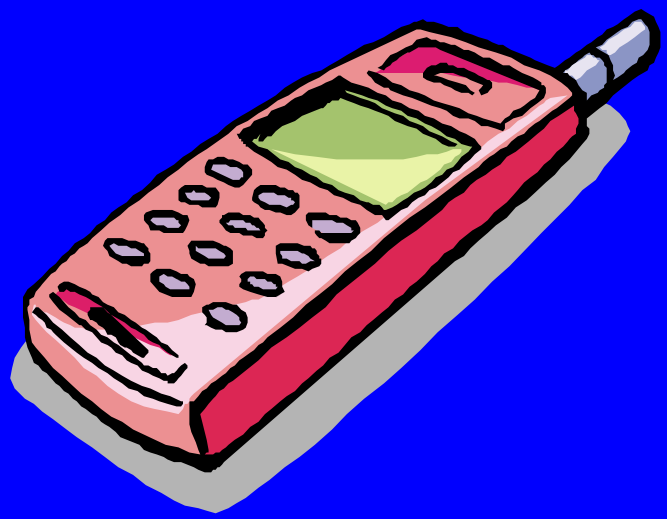
Handset Glitches are Real

- **“NTT DoCoMo issued a recall and halted sales of its new videophone on Monday after discovering a software glitch that wipes out most of the data stored inside the handset.”**

- APRIL 06, 2001 (COMPUTERWORLD) - Nokia Corp. may have to recall millions of cell phones due to a software glitch that renders them inoperable with the third-generation (3G) networks that major U.S. carriers plan to start using this year.

Handset manufacture View

- Stimulate sales of new handsets
  - Excite user with new capabilities
- Improve competitive position
  - Introduce innovative features
  - Need to get handsets out faster
- Need to reduce liability for firmware bugs
- Ability to offer upgrades

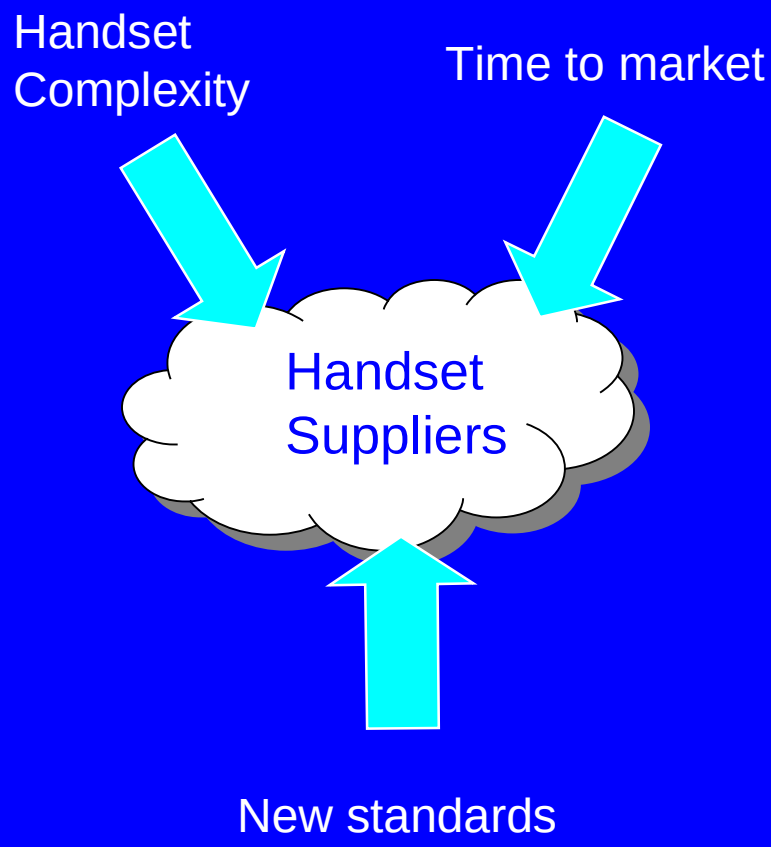


Operator view

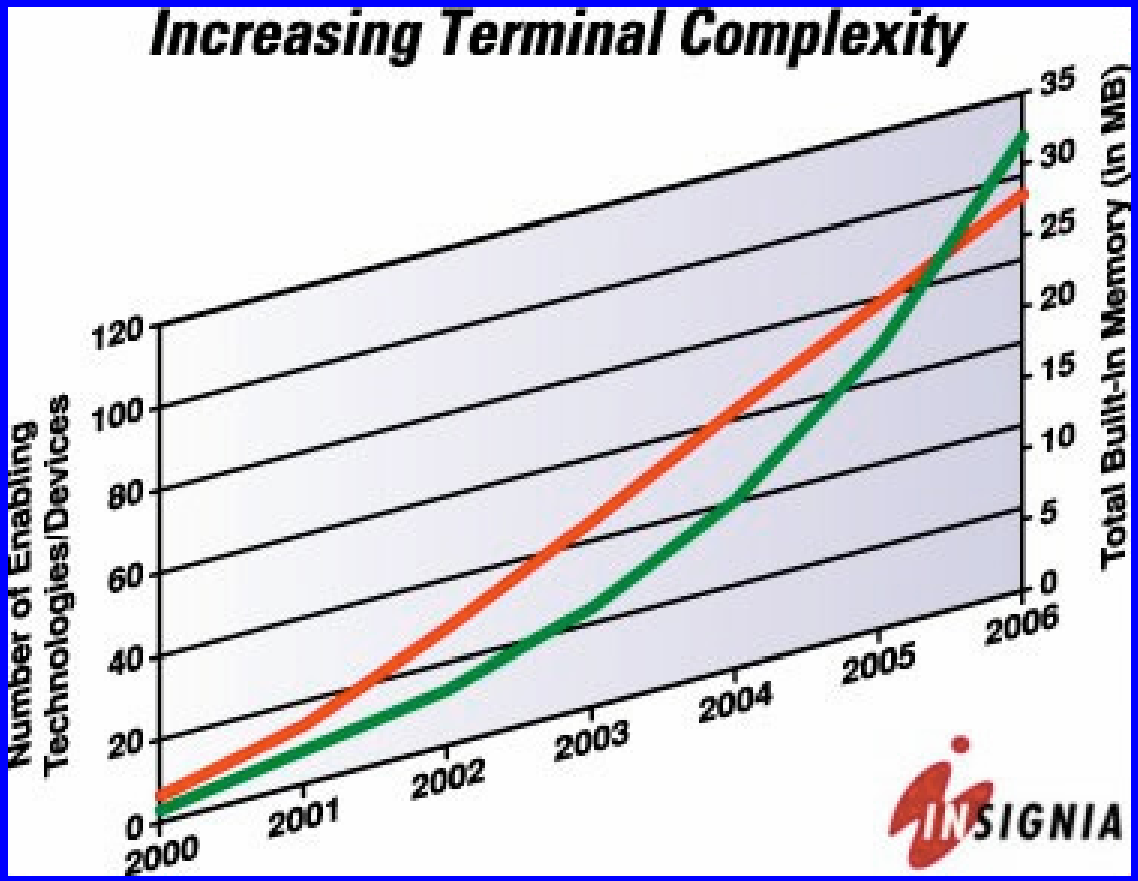
- Need to drive higher ARPU
- Desire to launch new services/features quickly
- Desire to capture the customer
  - Support a proprietary “platform”
  - Need to provide good customer service
- Desire to sell specialized services and features
- Minimal impact to customer and operator for bugs

The Problem

- Handset complexity increasing
  - Size of memory increasing to 32-64MB (2-4MB in 2002)
  - 3G technology driving advance capabilities
- Time to market shortening
  - Launch of new services and capabilities key to both handset manufacturers and operators
- New standards - Less experience with software



Handset Memory Size



Courtesy of Insignia

Environment
Why is this different?

- Limited access
  - No physical access
- Limited update resources
  - Limited program footprint
  - Limited storage memory for download
- Maintain handset sanity
- Limited bandwidth channel
- Unreliable channel

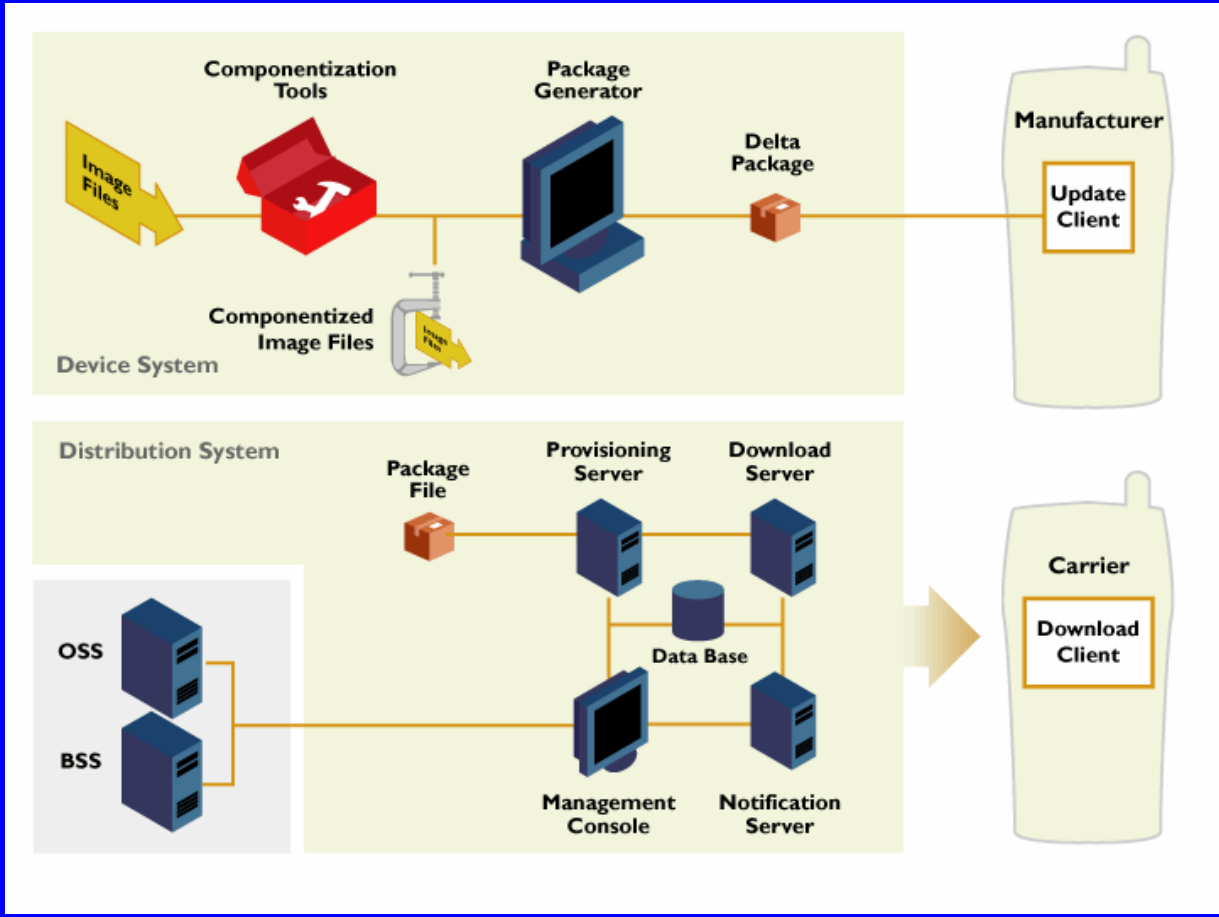
Key Challenges

- Can’t fail - leave handset inoperable
- Recovery from battery failure
- Recovery from communication failure
- Time to download
- Guarantee correct download
- Time to update
- Check for correct current version

The Concept

- Diff functions
  - Old version
  - New version
  - Compute compressed diff
- Diff creation and management
- Distribution system
- Handset client

System Architecture Example

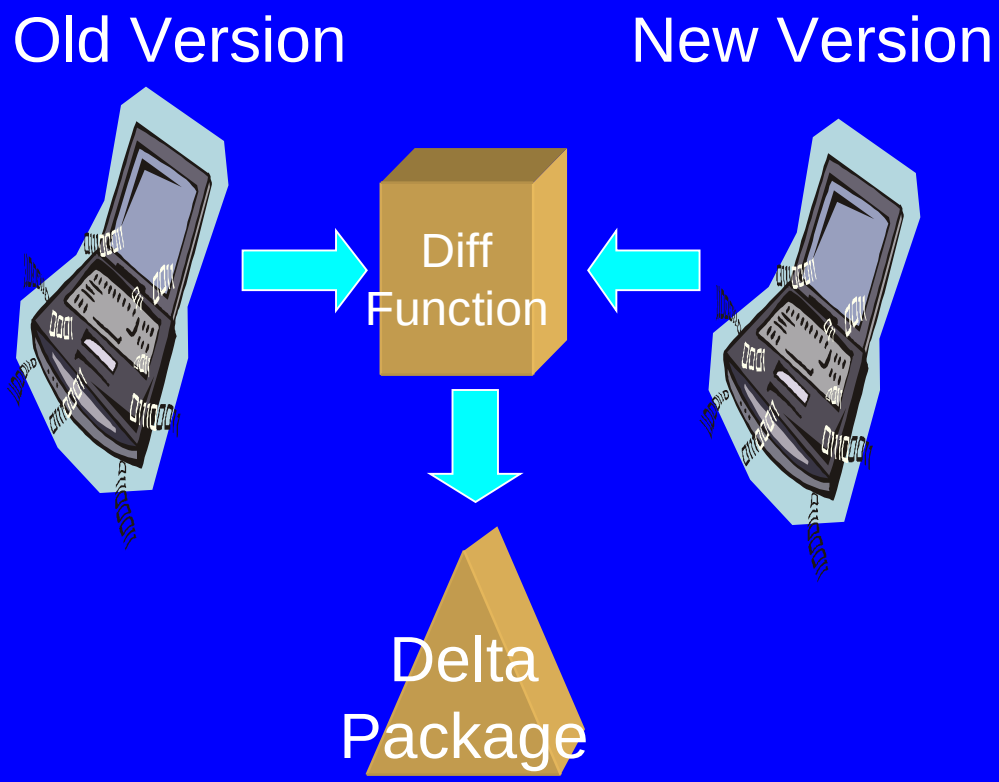


Courtesy of DoOnGo Technologies

Diff Computation

- Compressing file update by representing file differences
- Most use proprietary diff algorithms to reduce file size
- Performance parameters
  - File size
  - Update time
- File size impacts download time and storage space
- Update time driven primarily by number of blocks to be updated
- In order to reduce update time, must initially optimize program organization

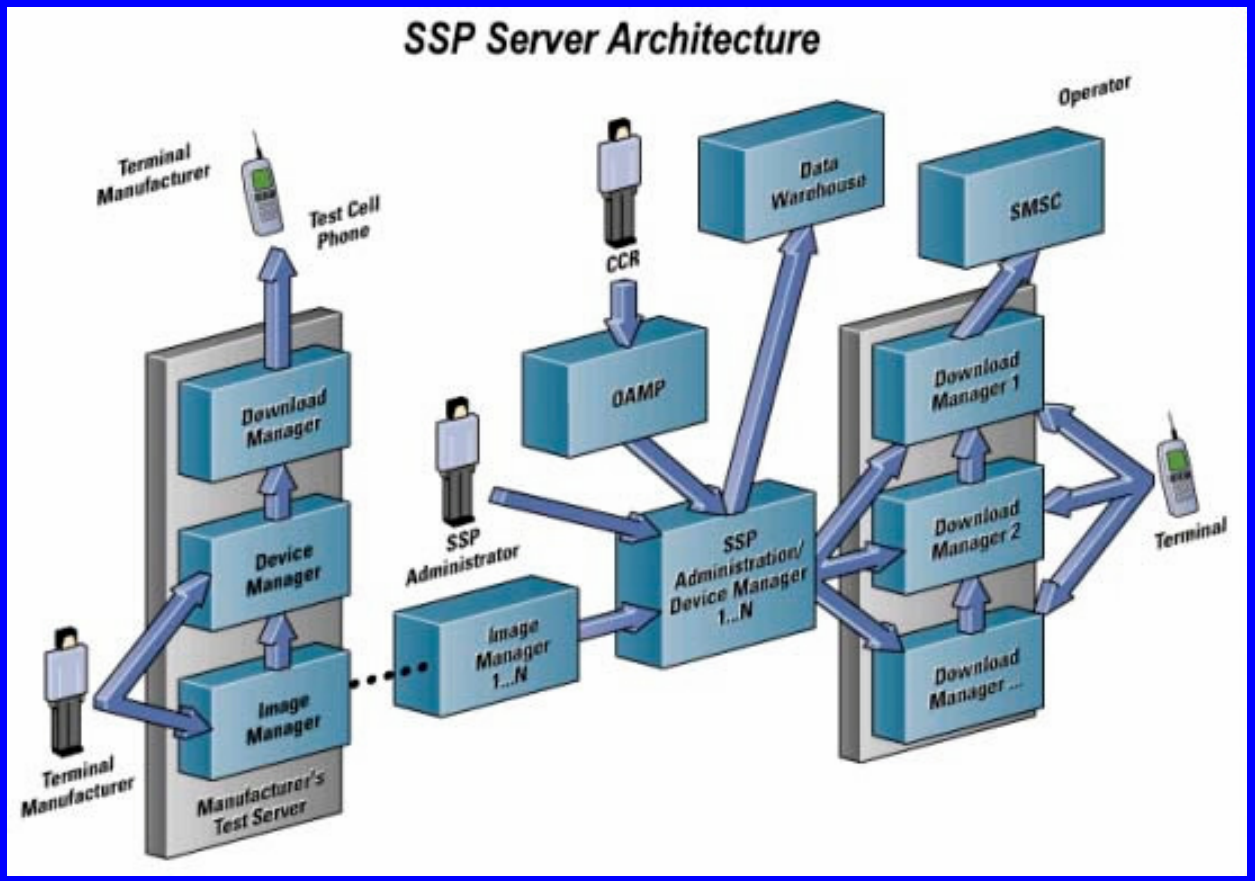
Diff Function



Distribution System

- Typically deployed by a carrier
- Also deployable by a handset manufacturer or other party
- Securely receive updates from manufacturer
- Manage user's on the system
- Send initiations to appropriate handsets
- Manage download to each handset
- Maintain records of interactions

Server Architecture Example

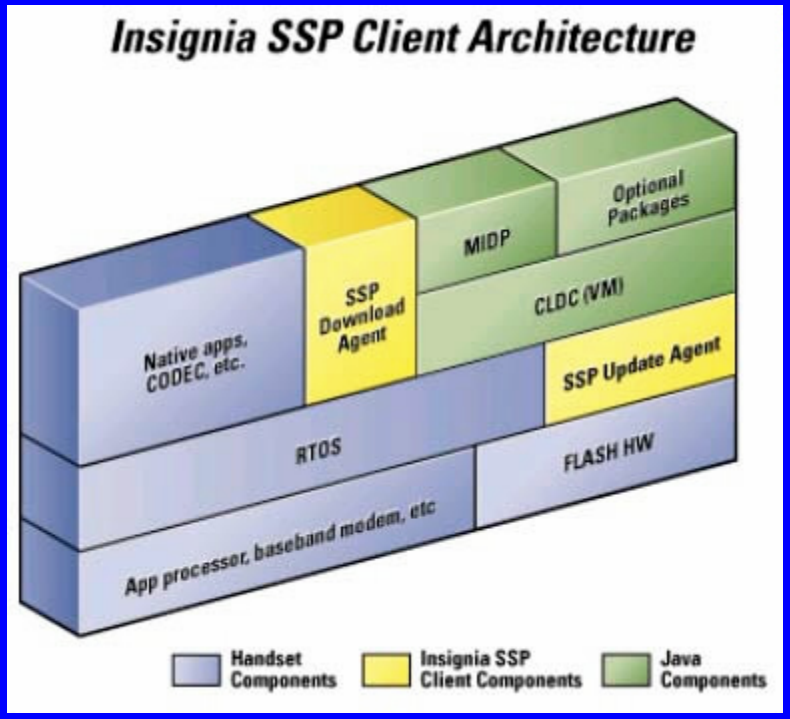


Courtesy of Insignia

Client

- Typically separate download and update clients
- Download client
  - Alerted by initiation
  - Identifies handset and software version
  - Interacts with user if needed
  - Securely downloads update
  - Stores update in memory
- Update client
  - Manages update of firmware
  - Unwraps update package (inverse diff)
  - Sequences the update of firmware, block by block
  - Manages restart of update if interrupted
  - Provides access to memory read/write functions

Client Architecture Example



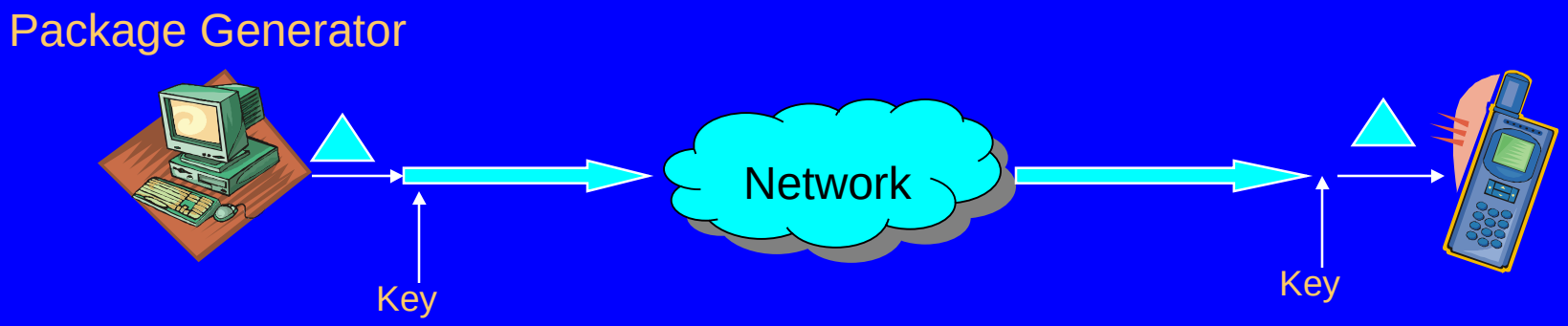
Typical Scenario

- Initiation sent to handset
- User is prompted with availability of update
- User accepts or rejects update
- Handset directed to source of update package
- Update package downloaded to handset
- Update initiated
- After update completed, handset returned to operational state

Courtesy of Bitfone

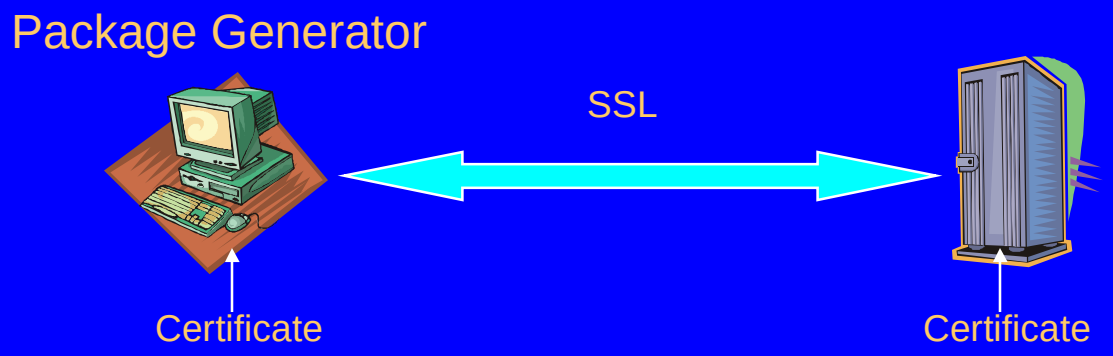
Security Issues

- Package authenticity and integrity
  - Need to validate that the package came from the trusted source (handset manufacturer).
  - Need to guarantee that the package had not been altered along the way.
  - Some form of digital signature is used.



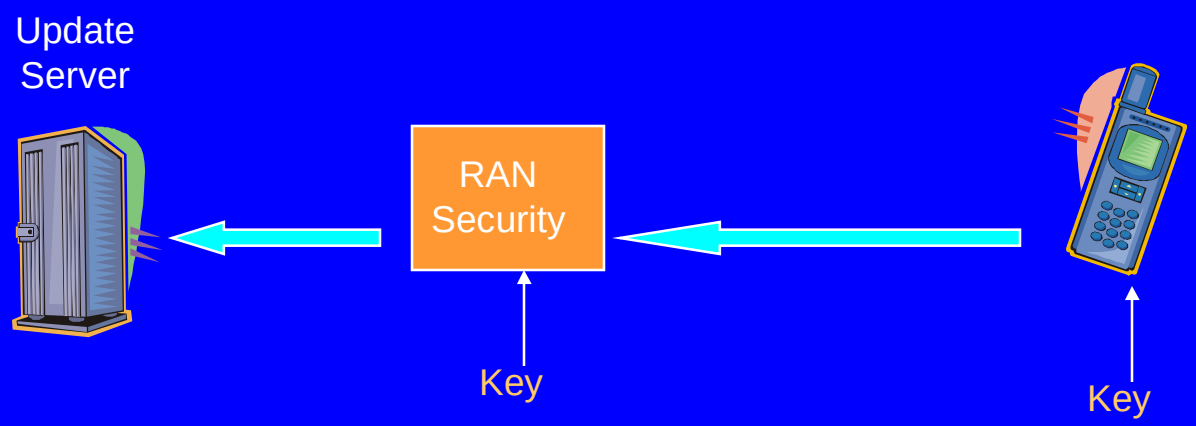
Security Issues (cont.)

- Package submission authentication and privacy
  - Need to make sure the server system is getting the packages from the correct source.
  - Need to make sure that the registered server system is authenticated.
  - Typically, SSL session between package generator and server.



Security Issues (cont.)

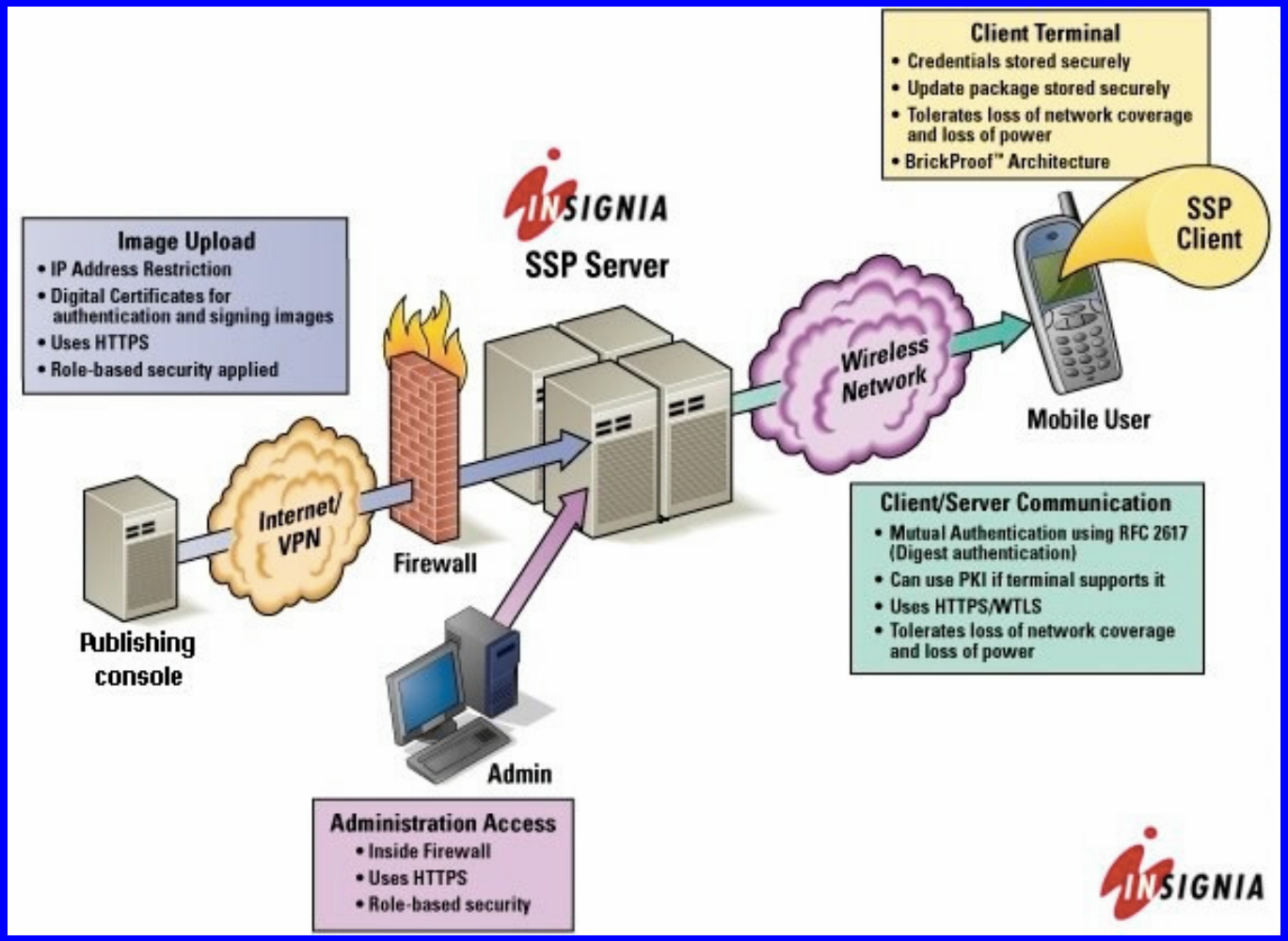
- User/device authentication
  - The device is the correct device
  - This is an authorized device for service in the netework
  - If a paid download, the user is authenticated and authorized for the download.



Security Issues (cont.)

- Server authentication
  - Make sure that the server is not an imposter
  - Avoid denial of service attack.
- Digest Authentication with key or password
- PKI Approach better if handset has this capability

Security Architecture

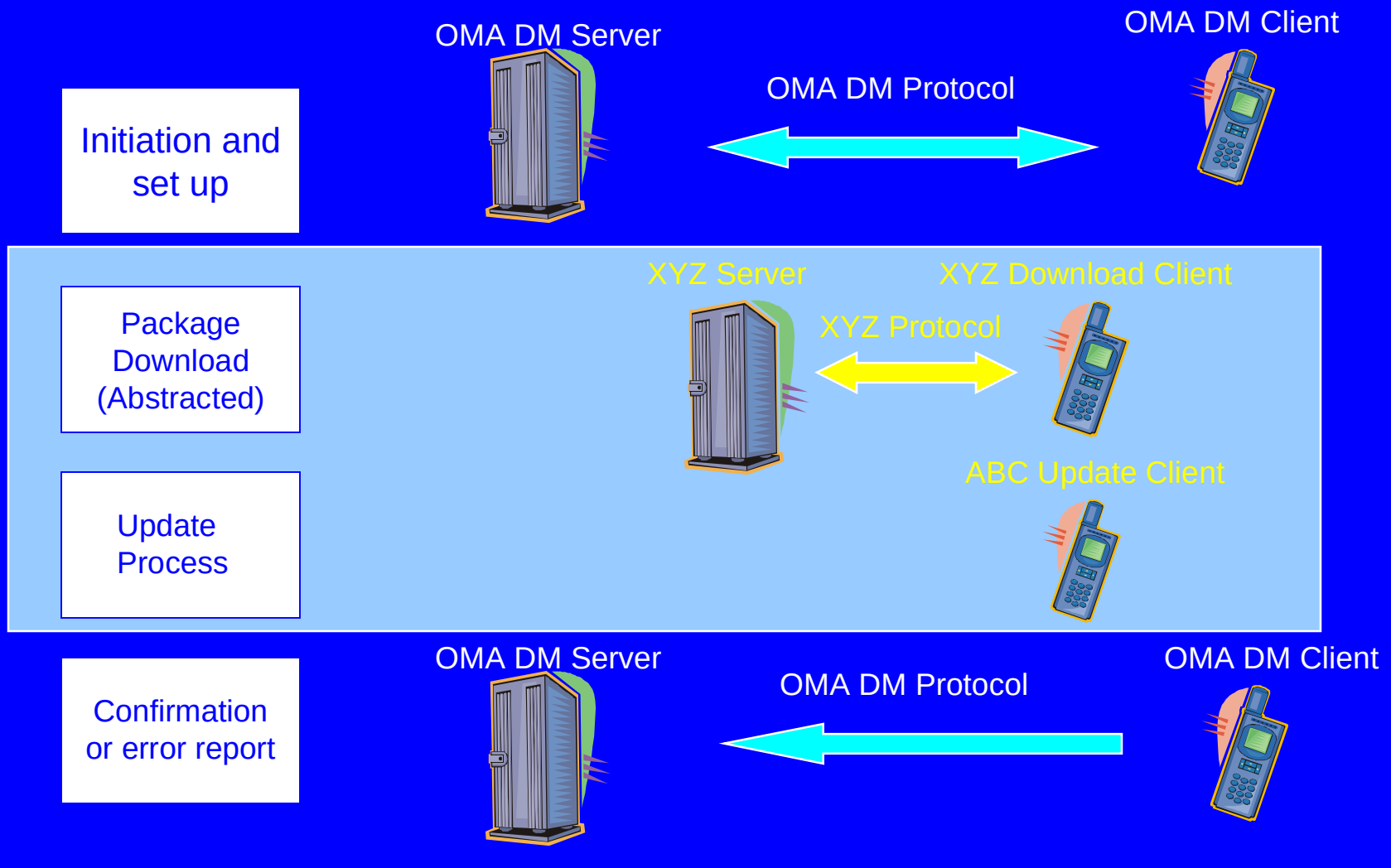


Courtesy of Insignia

Standards Activity

- Open Mobile Alliance is the main standards activity
- Part of the OMA-DM (Device Management) working group - Formerly called SymcML-DM
- Firmware Update is a capability being defined within the OMA-DM standard
- OMA-DM defines a data structure called a Management Tree and an XML syntax for primitives that operate on these data elements.
- The firmware update standard defines:
  - The data objects in the management tree specifically for firmware updates
  - The SyncML commands that are used to:
    - Copy data between server and client
    - Initiate the download and the update

OMA Structure (Tentative)



Directions of Technology

- Main suppliers today - several small companies provide client and server technology
- Some handset manufacturers doing their own
- Focus is on bug fixes and version updates now
- Feature addition later
- Perhaps application download and management
- Other handset management functions

Who Will Operate These Systems?

- Carriers initially
- Handset manufacturers perhaps
  - Puts control and responsibility on HS Manufacturers
  - Who pays for air time
- Third party distribution
  - Security greater issue
  - Good for paid downloads
- Free or paid updates

The Future

- Will this technology continue to be supplied by the smaller companies?
- Will handset manufacturers do their own or buy this technology?
- Will the server systems accommodate divergent client suppliers?
- Will the standards set the direction or will the technology set the standards?
- Will this function be combined with other device management functions?
- What device management functions?