



Understanding the IMS in JME enabled devices, JSR-281

Piotr Kessler
Chief Architect Client SW

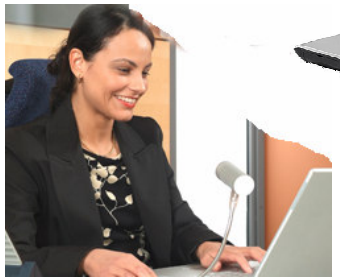
ERICSSON
TAKING YOU FORWARD

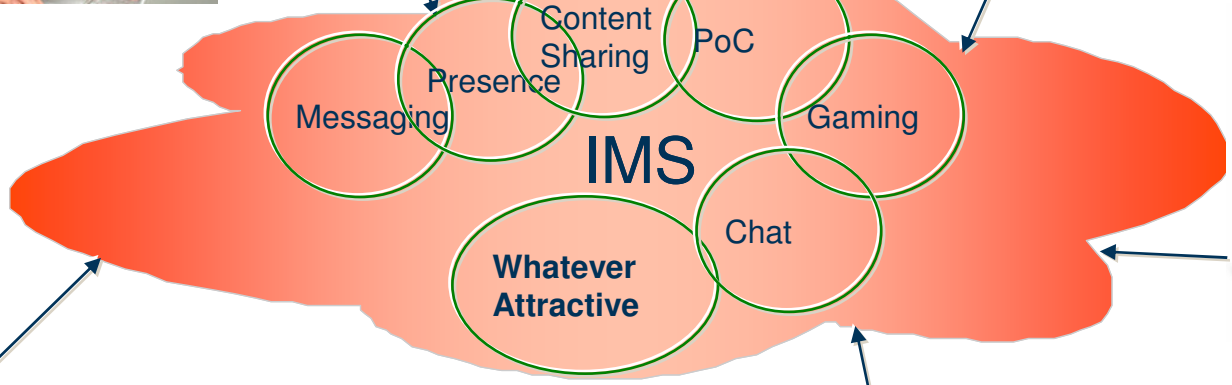
Agenda

- **IMS Introduction**
- Realization of IMS services in Devices
- JSR-281 – standardized way forward
- Ericsson ICP & example of services
- E2e service creation

It’s not just about IMS...

- it´s about people enjoying fancy services in their devices





IMS...

- it´s about merging technologies to easy deliver services

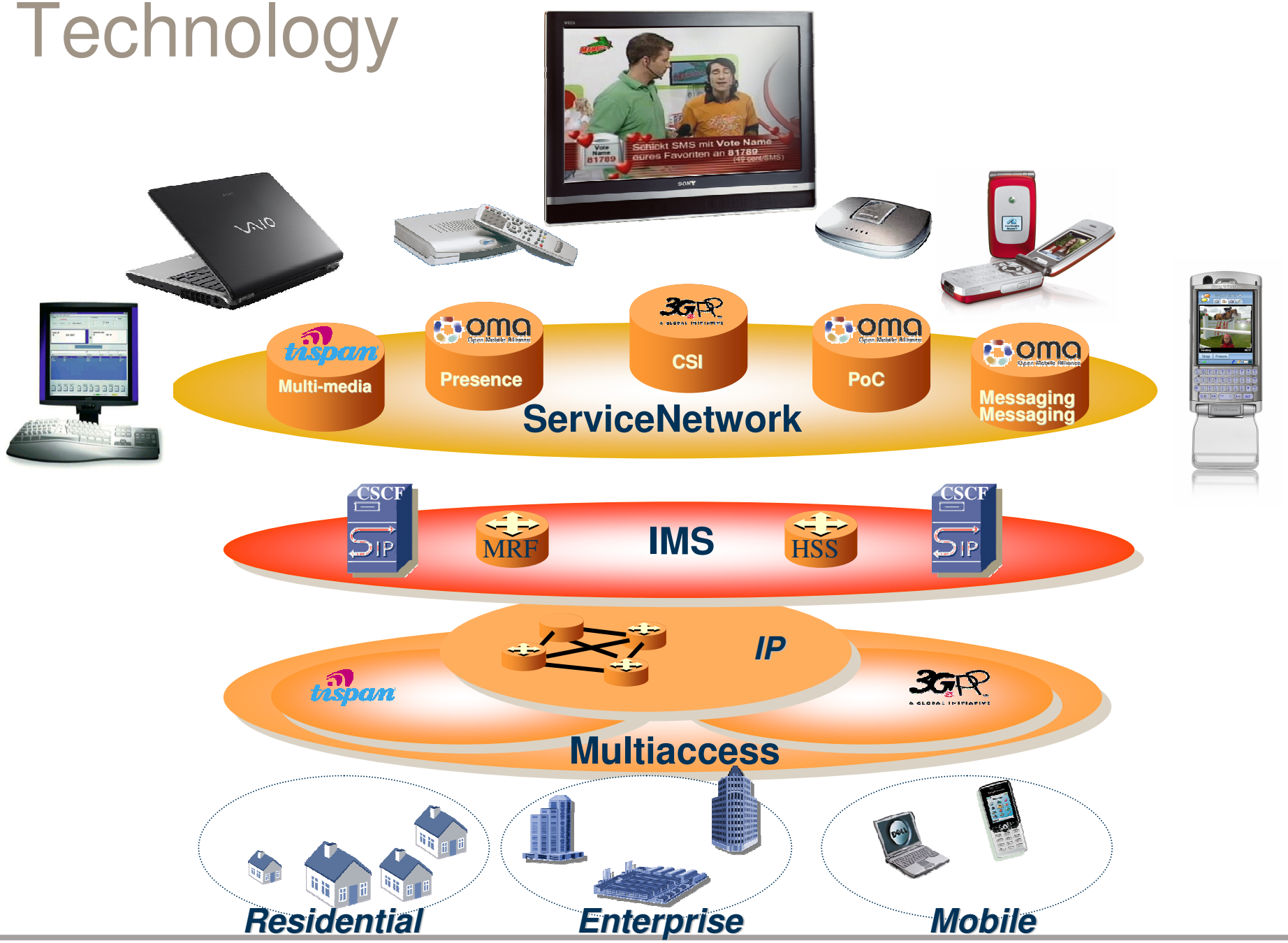
Messaging
Presence
Content Sharing
PoC
Gaming
Chat
Whatever Attractive

- Openness
- Innovation
- Short TTM
- Multimedia

IMS

- Standardization (3GPP and IETF, ITU-T, ANSI, ATIS, TISPAN, OMA, GSMA)
- Interoperability
- Multiaccess
- Secure reachability
- Payment security
- Quality of Service

Technology


Multi-media
Presence
CSI
PoC
Messaging
Messaging
ServiceNetwork
CSCF
MRF
IMS
HSS
CSCF
IP
Multiaccess
Residential
Enterprise
Mobile

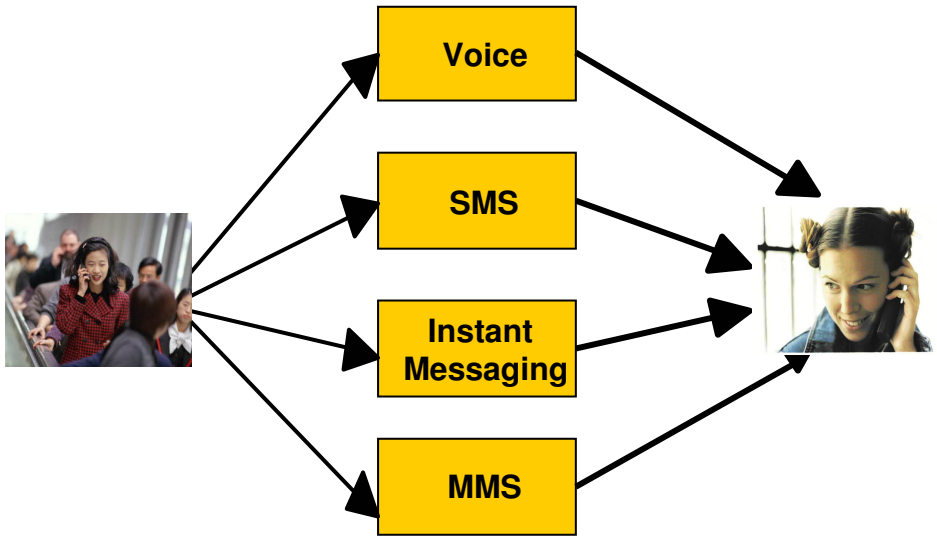
IMS is the framework

- E2E
- Reachability
- Mobility
- Interoperability
- Convergence
- Quality of Service
- MultiMedia connections
- Security
- Charging

IMS Vision - User perspective

Pre- IMS Communication

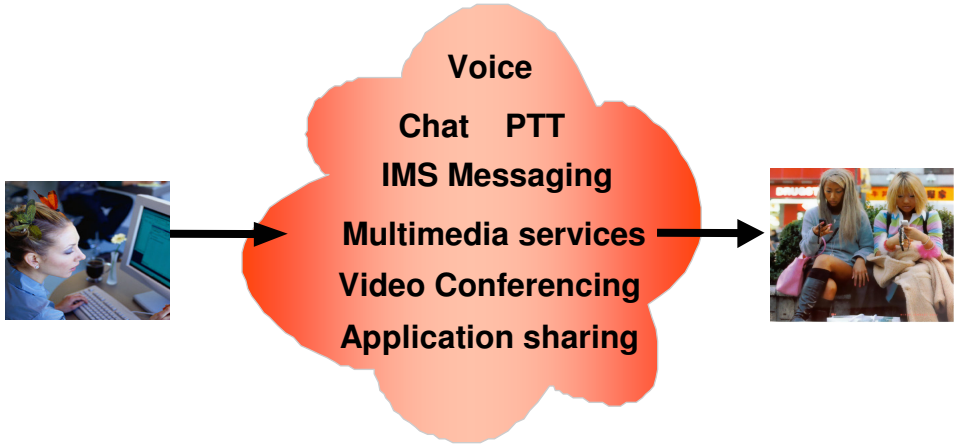
Different communication applications



1. **Choice of communication type**
2. **Start of session**
3. **End of session**

IMS Supported Communication

Integrated communication application



1. **Choice of initial mode with presence**
2. **Start of session**
3. **Dynamic change of media within the session**

To build the “best” user experience

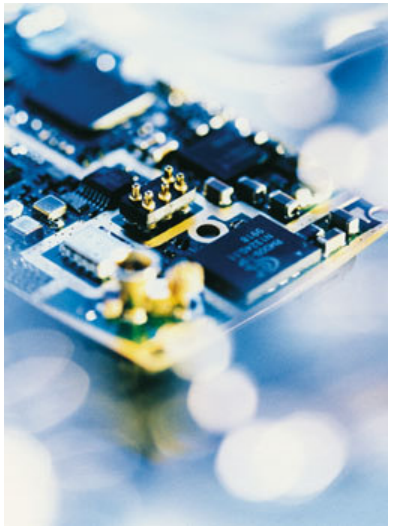
Who | **How** | **Add participants** | **Add media**

Agenda

- IMS Introduction
- **Realization of IMS services in Devices**
- JSR-281 – standardized way forward
- Ericsson ICP & example of services
- E2e service creation

Developer's perspectives – a challenge

- Technology
  - A number of communication protocols to use SIP/SDP/RTP/RTCP/MSRP/XML/XCAP…
  - A number of standards to follow
  - A number of Real Time requirements to fullfill
  - A number of low-level tasks to manage
  - A number of service enablers to implement



- End-user rich application
  - A number of services to aggregate using service enablers
  - Attractive and competitive GUI for the user
  - Simple and engaging user interaction

But…

…the solution is here – two domains

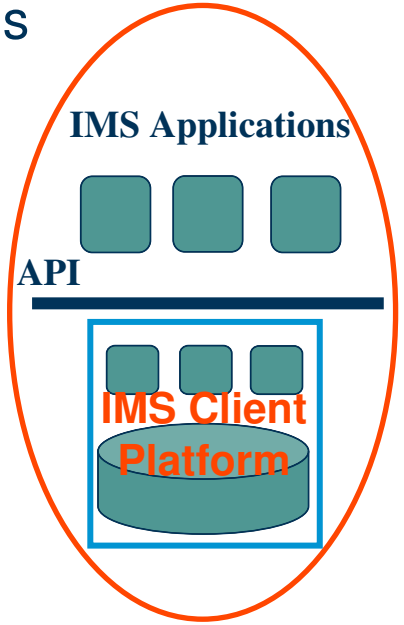
- IMS Client Framework
  - Domain of the device platform provider

- IMS application
  - Domain of the application developers

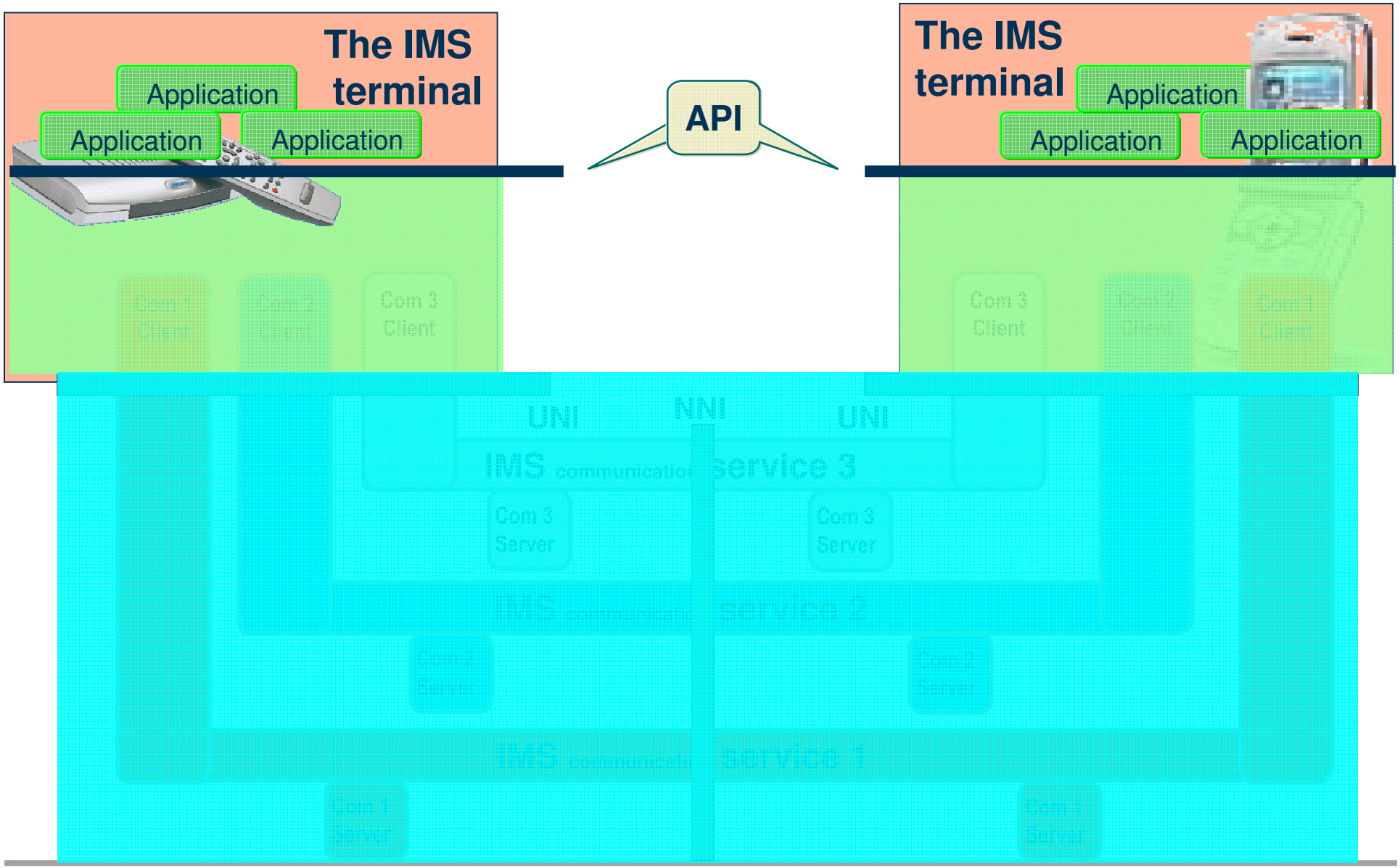
A developer-friendly IMS terminal

- Layered architecture
  - IMS Client Platform – realization of IMS Framework
    - Focus on IMS technology
    - Focus on service logic
    - Focus on co-location of Applications
  - High-level API
    - Hides all above details
    - Offers straight-forward application-level methods
  - Applications
    - Focus on usability and user interaction
    - Focus on attractive GUI
    - Focus on attractive service behavior
    - Focus on fast TTM

API

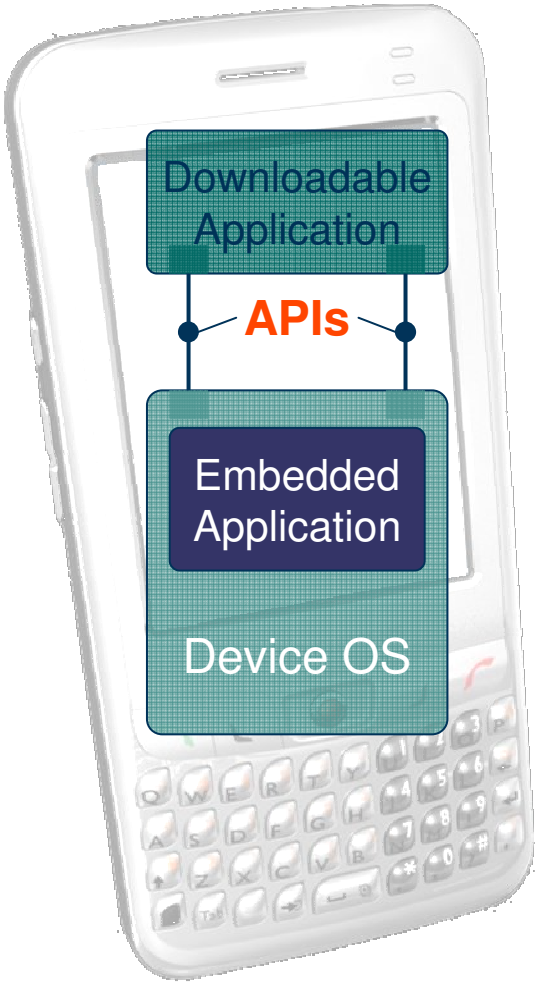


IMS E2E concept



Types of IMS Applications

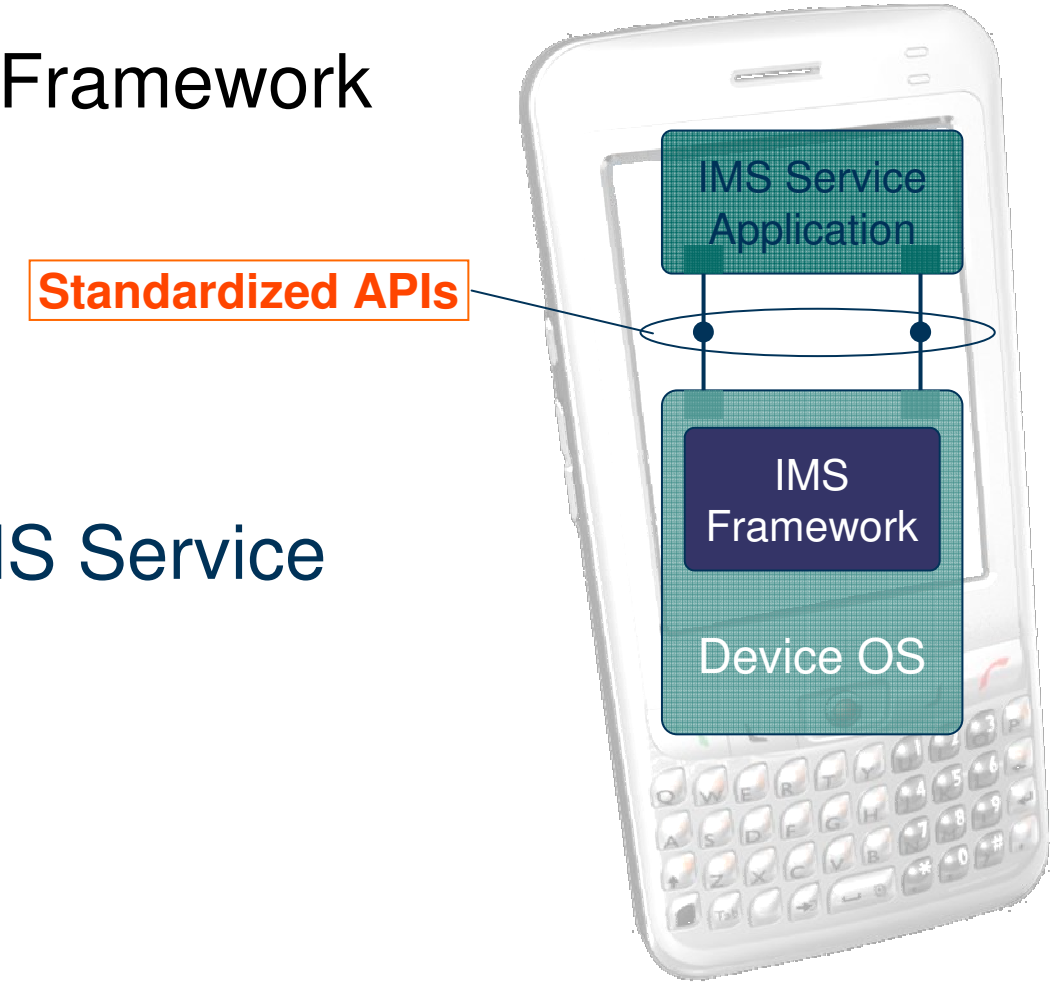
- ‘Embedded’ Applications
  - Applications are pre-installed into the device
  - Tight integration with the device's Operating System (OS)
- Downloadable Applications
  - Applications are installed after device leaves the manufacturing process
  - Required: Open and – preferably – standardized Application Programming Interfaces (APIs)



Merging both advantages

- ‘Embedded’ IMS Framework

- Downloadable IMS Service applications



Agenda

- IMS Introduction
- Realization of IMS services in Devices
- **JSR-281 – standardized way forward**
- Ericsson ICP & example of services
- E2e service creation

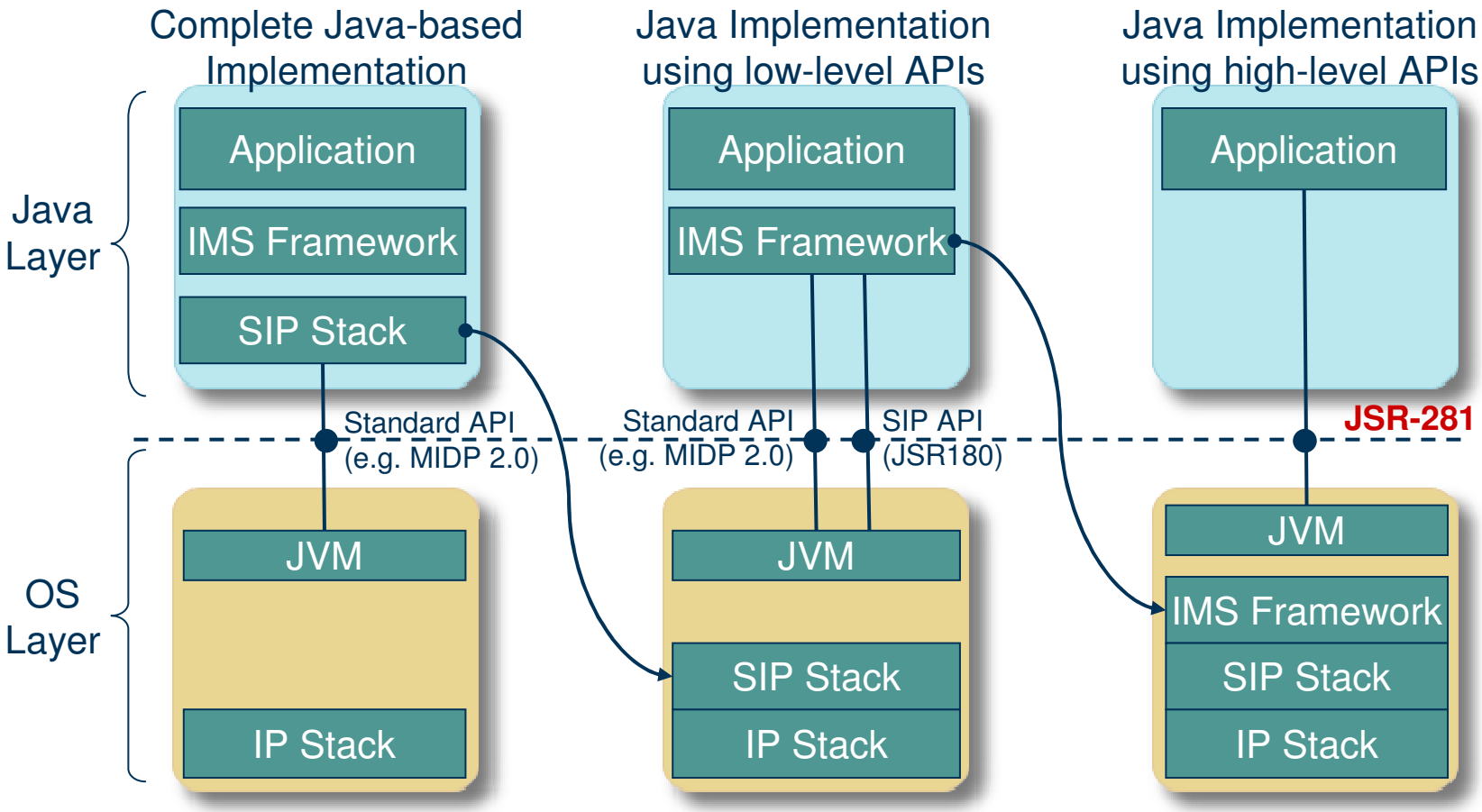
Standardized way forward: JSR-281

IMS Services API

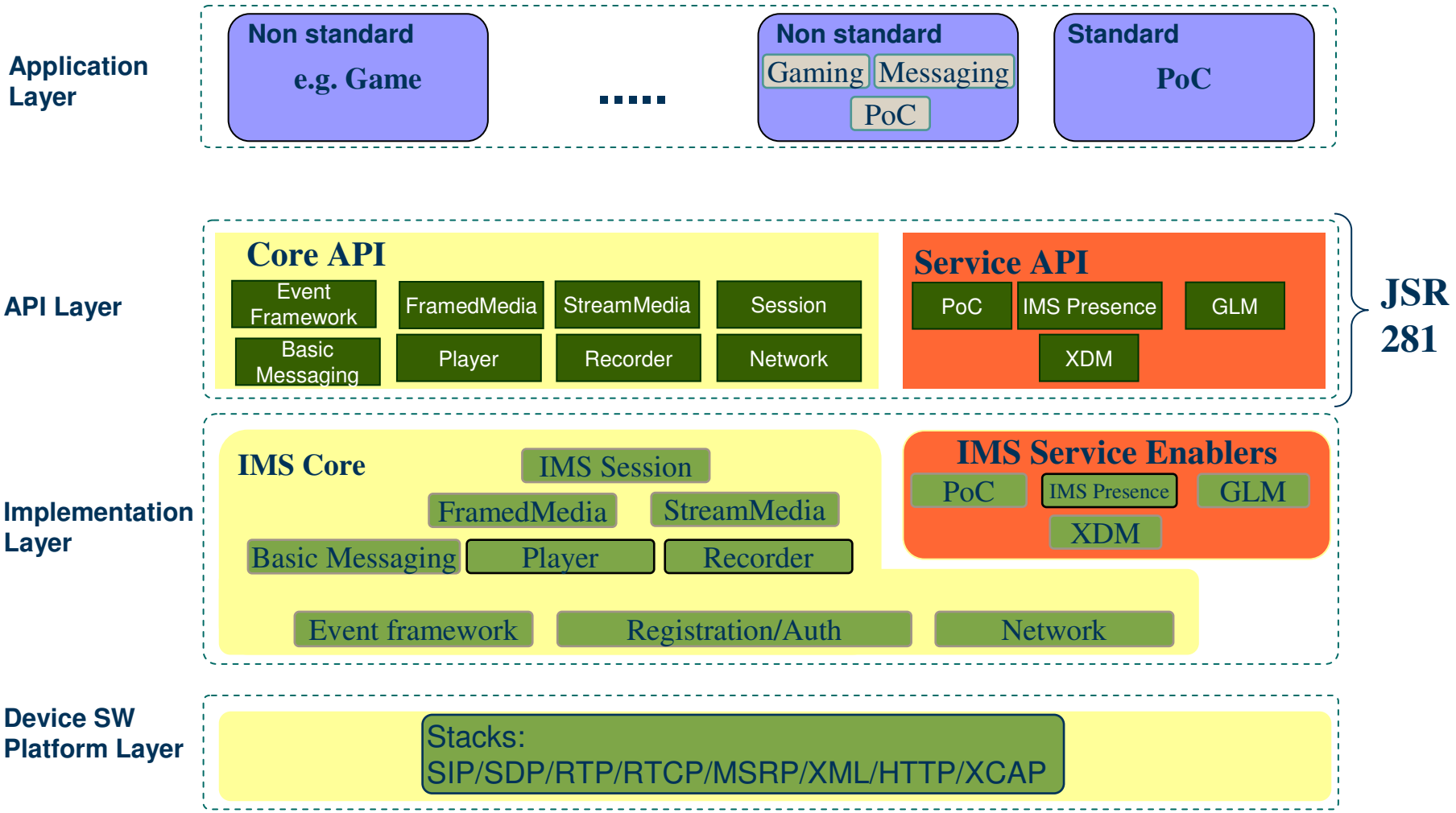
- API for Client application development for JME devices: CDC/CLDC
- Abstracts IMS technology through API
  - Generic IMS API
  - IMS Services API
- Brings standardized IMS Client Service Creation toolbox for Java Development Community
- Lead by:
  - Ericsson (Piotr Kessler and Stefan Svenberg)
  - BenQ (Volker Bauche and Mirko Naumann)

JSR-281 EG members

IMS functionality in client applications - the way to JSR-281

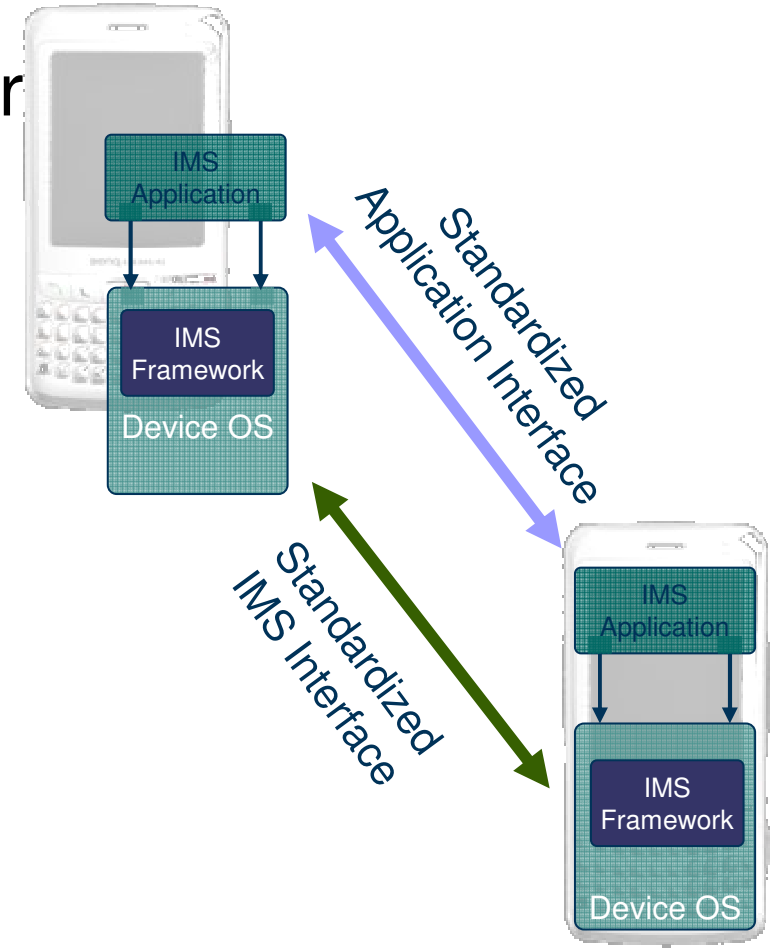


JSR-281 architectural concept



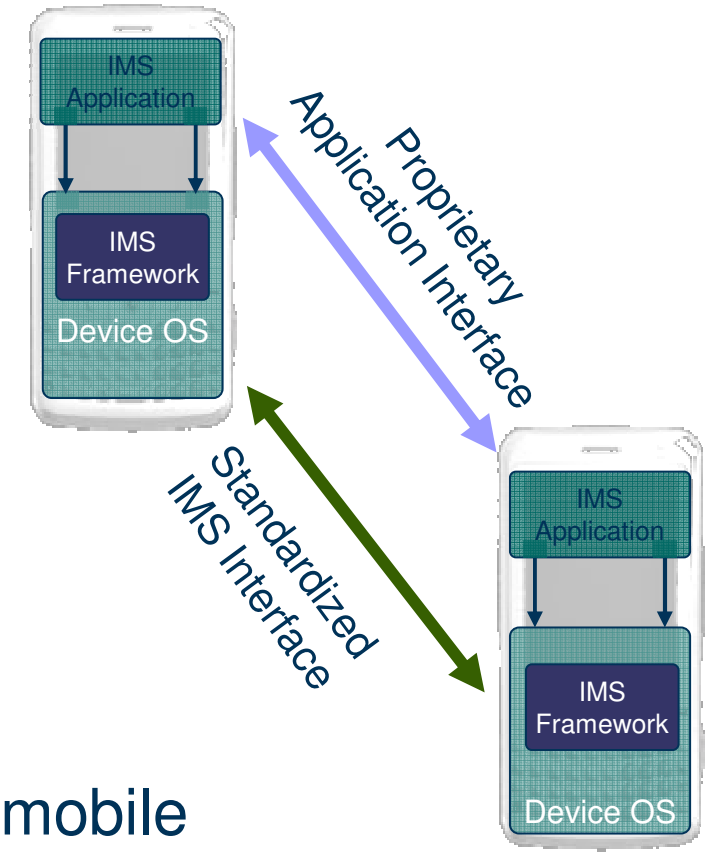
Standardized IMS Services

- All aspects of the service ar standardized
  - User-related features
  - Architecture
  - Protocol handling
  - Application level interface
- Example
  - OMA PoC

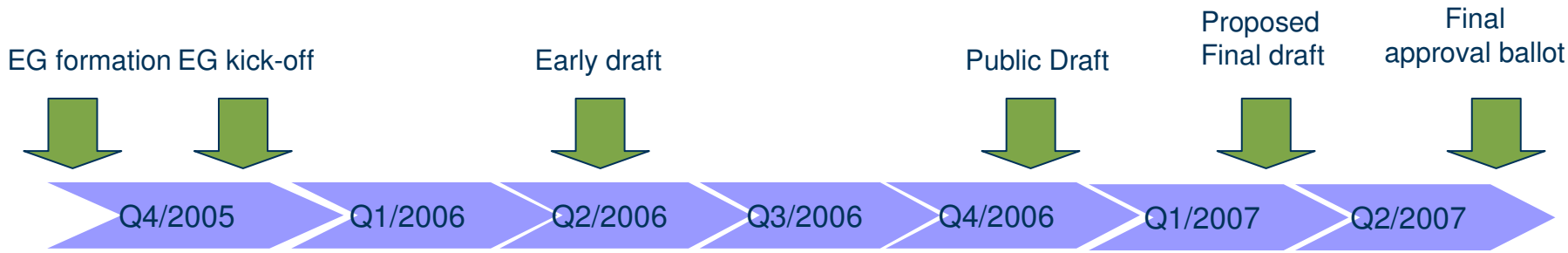


‘Non Standardized’ IMS Services

- Combination of standardized key IMS service enablers
  - PoC
  - Instant Messaging
  - Group & List Management
- and application defined multimedia sessions
- Application level interfaces 'application-specific‘
  - A.k.a. 'proprietary,
  - E.g. user plane from mobile to mobile



JSR-281 main milestones

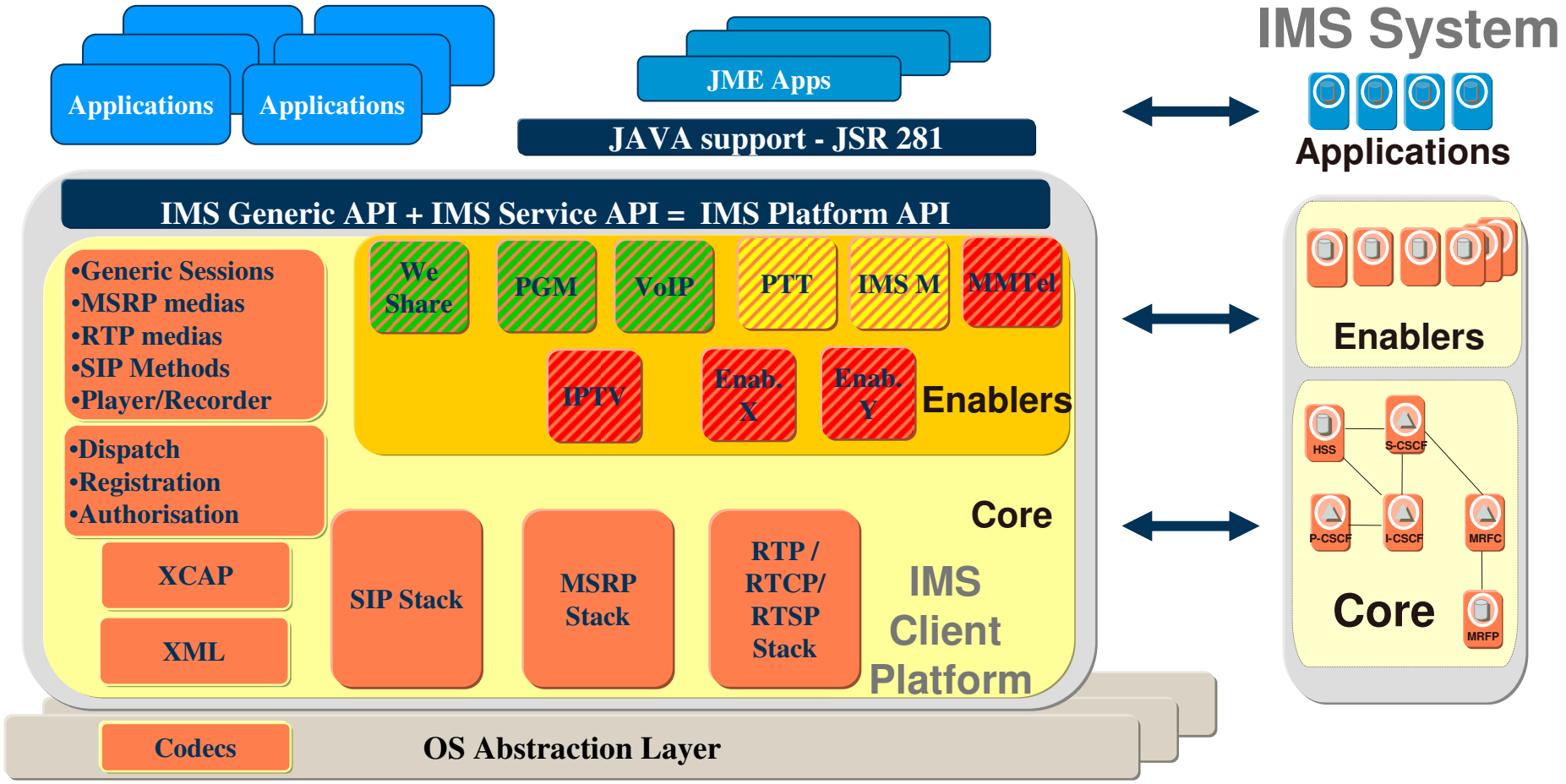


- Early Draft: Q2 2006
  - Requirements & Spec developed for first public review
  - Reference Implementation work started
- Proposed final draft: Q4 2006
  - Specification ready
  - RI and TCK assumed ready, still possibility to finalize
- Final Approval Ballot: Q2 2007
  - TCK & RI made available for licensing

Agenda

- IMS Introduction
- Realization of IMS services in Devices
- JSR-281 – standardized way forward
- **Ericsson ICP & example of services**
- E2e service creation

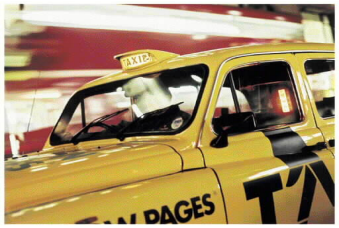
Ericsson IMS Client Platform Architecture



Ericsson IMS weShare

weShare Image

Send a picture during an ongoing conversation



weShare Motion (GSMA Video Share)

Send a live video during an ongoing conversation

M600

weShare Media File

Send pre-stored information during an ongoing conversation, i.e. picture, e-mail, video clip, film

weShare Whiteboard

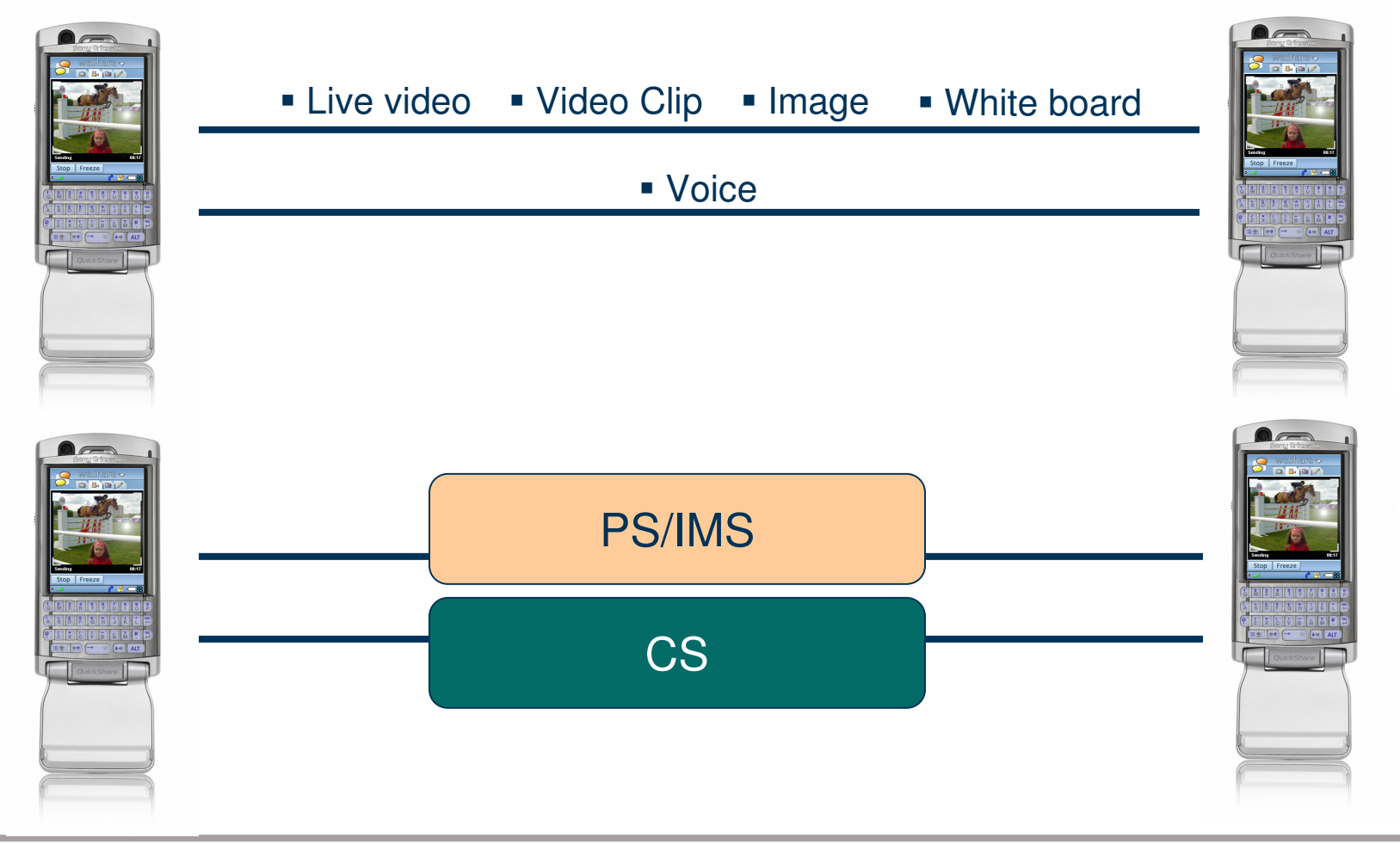
Share a whiteboard session during an ongoing

W950

Future application: weShare Game, weShare Music, weShare Web etc.

weShare Communication Principle

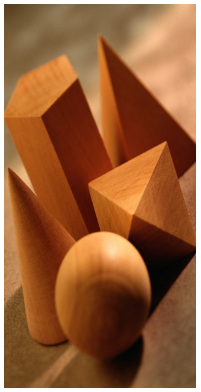
- A combinational service is created by combining a CS speech call and one or more PS media streams.



Agenda

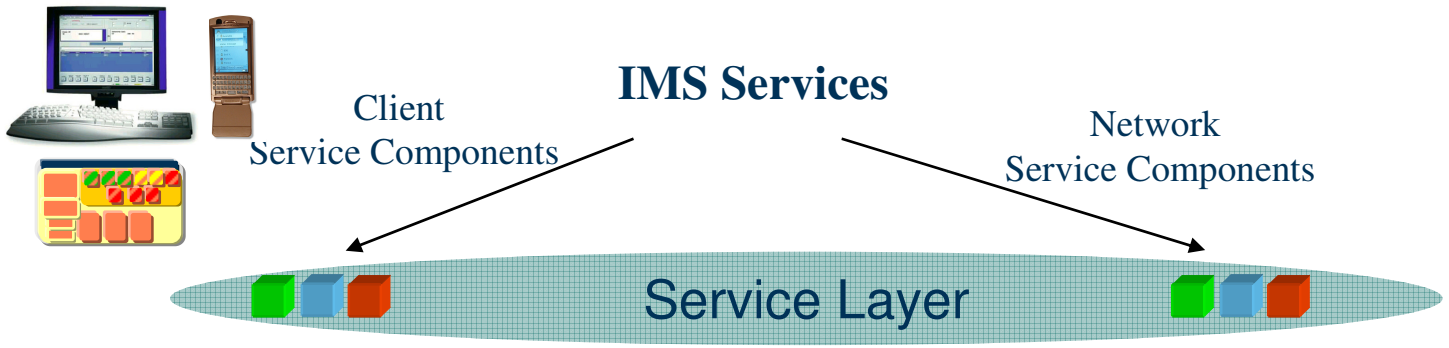
- IMS Introduction
- Realization of IMS services in Devices
- Ericsson ICP & example of services
- JSR-281 – standardized way forward
- **E2e service creation**

IMS Service Creation Solution – Service Development Studio



- Eclipse based Service Development Studio: use of existing tools and skills
- Target Developer: Internet-savvy developer with Java ME, Java SE and Java EE experience
- Server side development: SIP Servlet engine with JSR116 API
- Client side development: IMS Client Platform with pre-JSR281 API for Java ME and Java SE
- Simulated IMS infrastructure + emulated device

E2E service creation for IMS



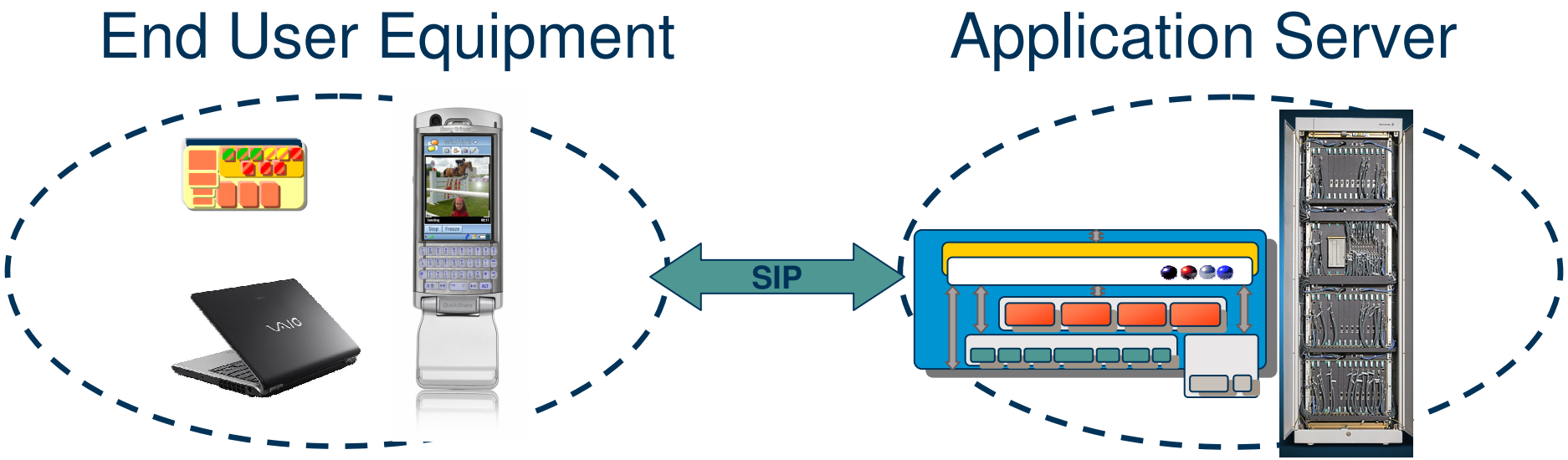
- Initial concerns:
  - TTM: efficient development & deployment of e2e services
    - High abstraction level of API
    - Verification on PC without access to real network
  - IOT: interoperable e2e services
    - Standardized interactions secured by the client platform

E2E Service Creation for IMS
Part 1: Design Time

Service Development Studio

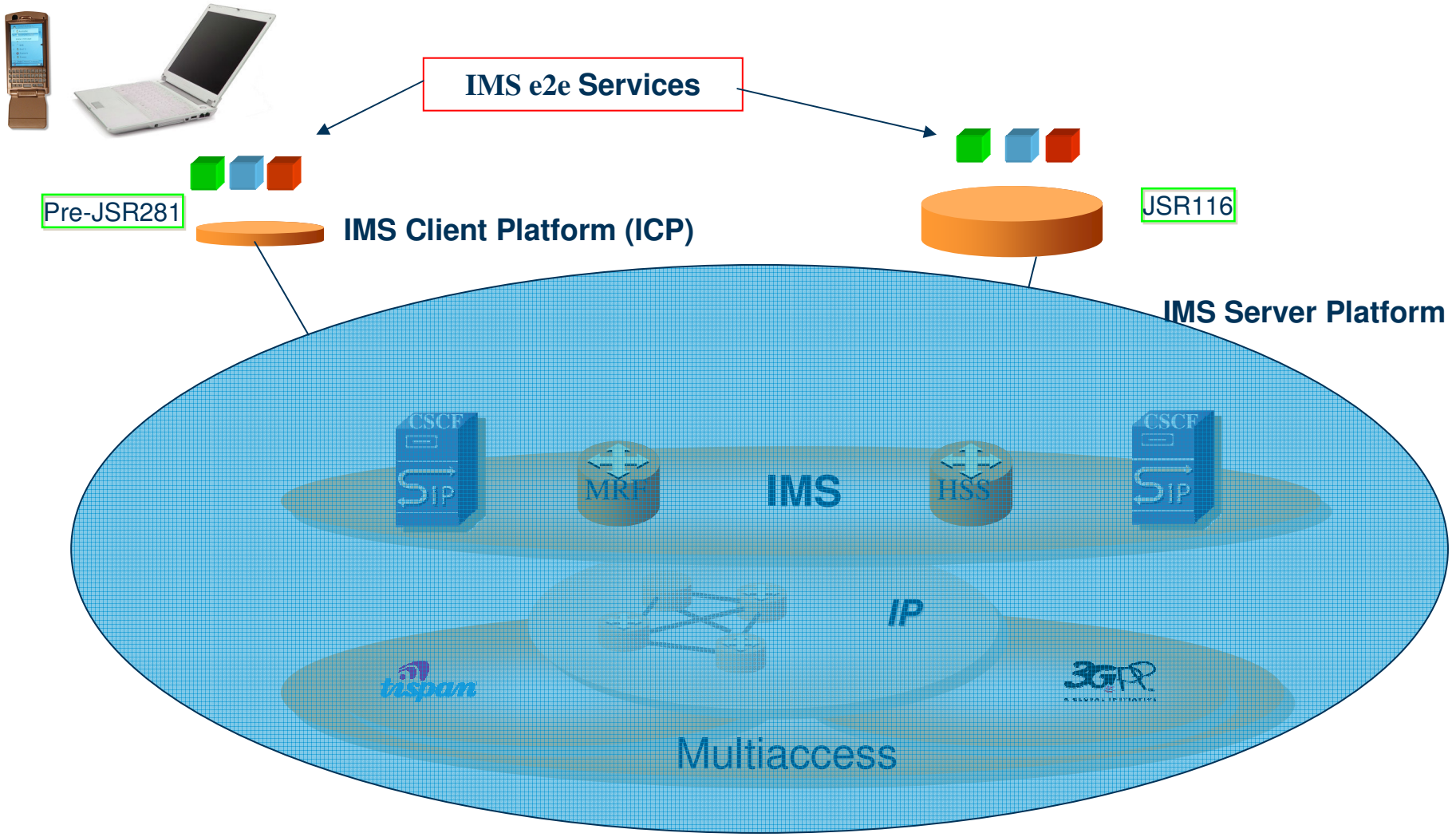
- For Linux/PC
  - Network - Simulation/Emulation
  - Client - Emulation

E2E Service Creation for IMS
Part 2: Deploy Time



- Installed IMS Platform
- Installed IMS Application
- Installed IMS Service

Abstracted view on IMS Network



Summary

- IMS provides framework for pervasive IP multimedia services
- IMS Client Platform enables simple client development…
- and secures convergence & interoperability
- JSR-281 brings everything together through standardized Java API

Q&A


ERICSSON
TAKING YOU FORWARD