



Parlay

An Overview

Drivers, History, Architecture, ...

Mehdi Khorasani

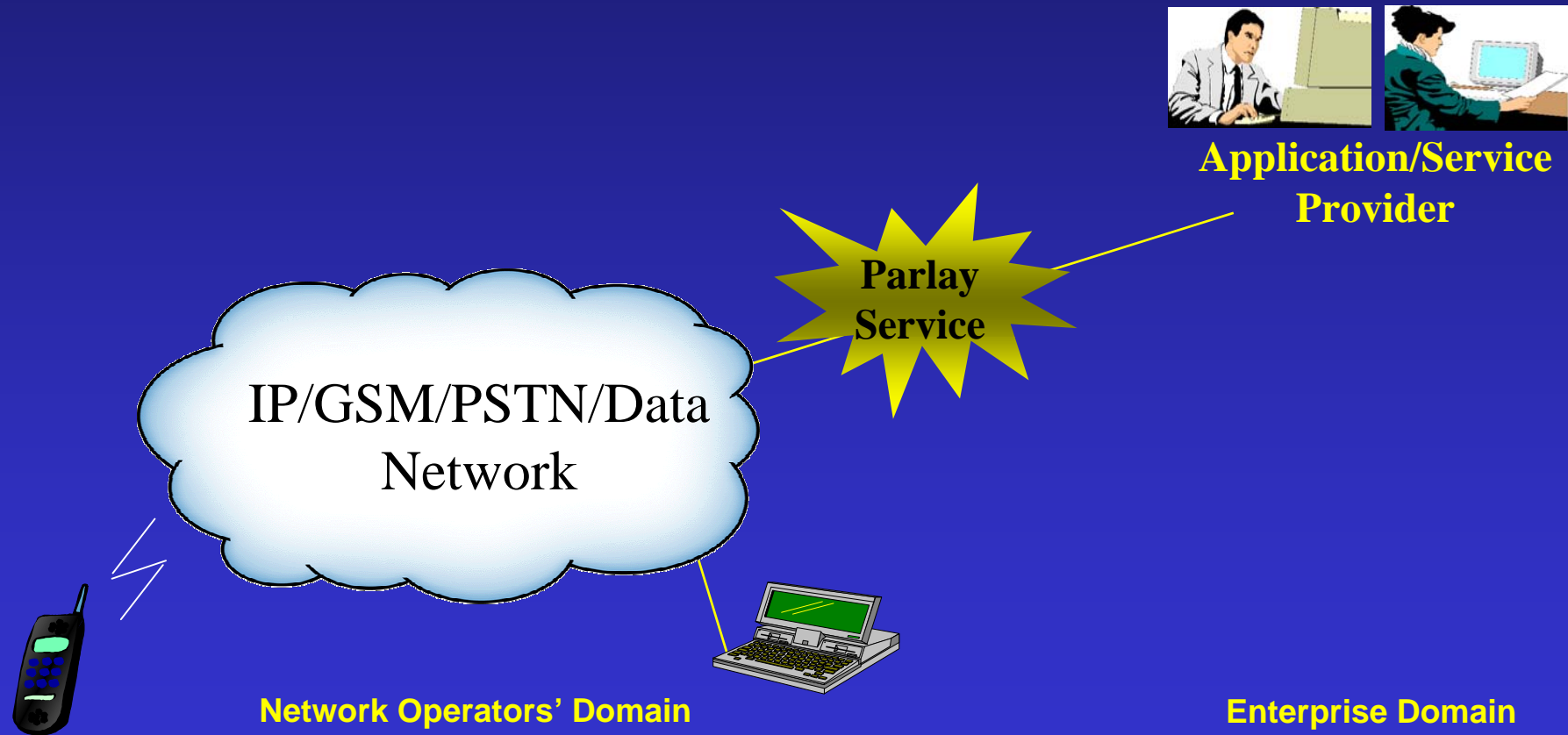
05 March 2002

Agenda ...

- *What is Parlay & Why?*
- *A Short History*
- *Parlay Architecture*
- *Parlay, JAIN, 3GPP, ETSI, ...*
- *Parlay Service Examples*

What is Parlay?

Is a set of Open Network APIs, which allows 3rd Parties to develop and run services external to operators' networks.



Why? Business & Technical Drivers

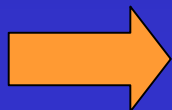
- Regulatory bodies are asking network operators to open up their networks to third party service providers
- Consequently, the number of service providers is increasing rapidly

-

The IT World
Rich in Applications
and Developers

Parlay APIs

The Converged Networks
Rich in Capabilities

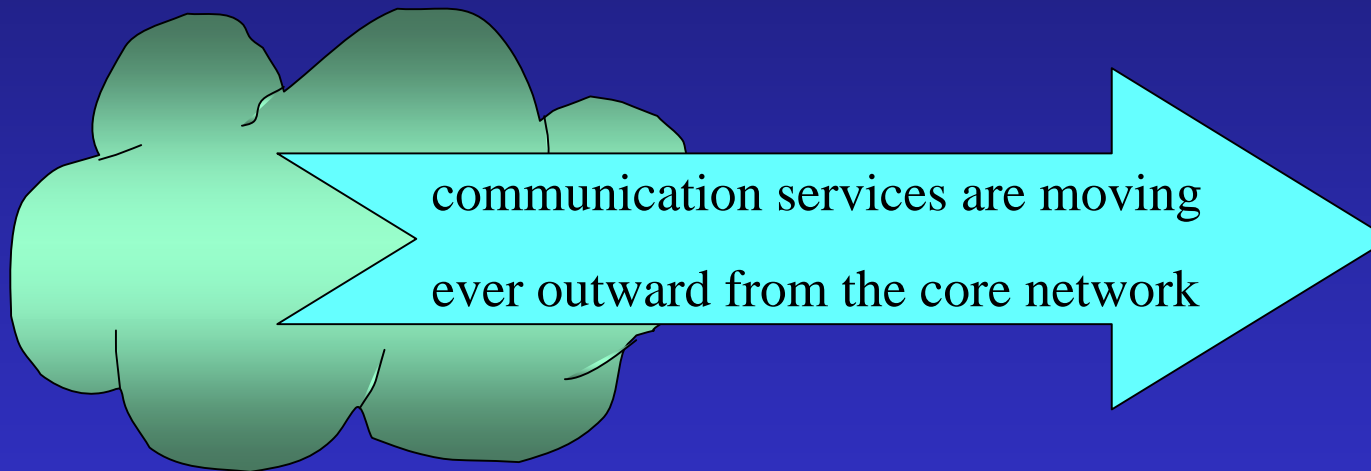


The model for delivery of communication services is moving towards that of a Service Provider Architecture.

Why? Business & Technical Drivers

- New Service Delivery Model:

A change in the network centric service delivery

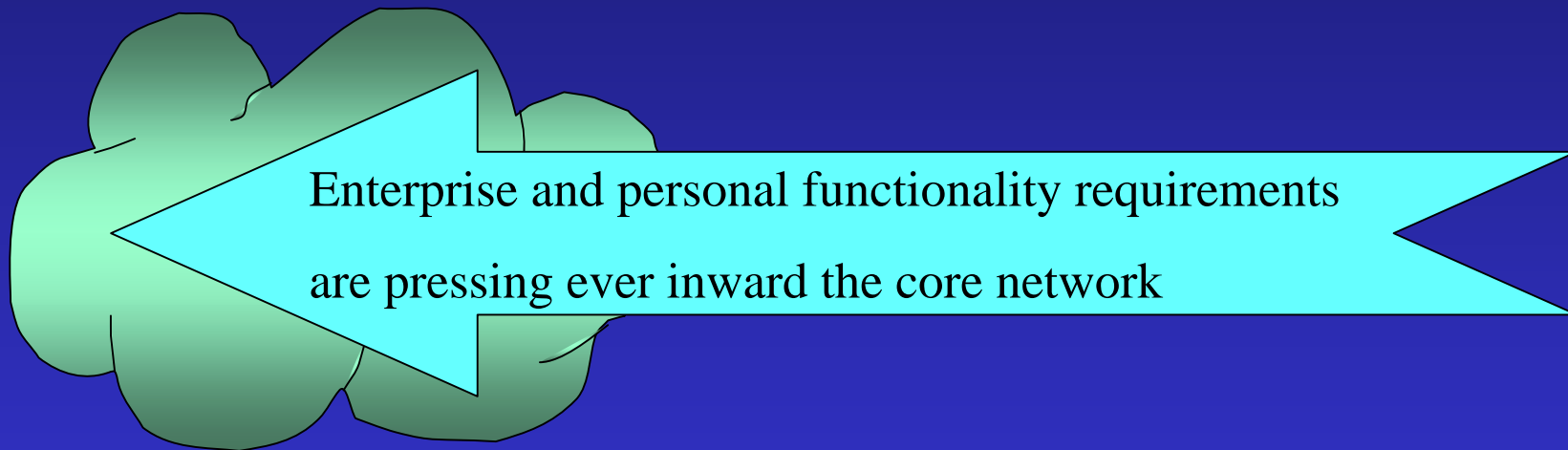


Examples are SCP based services (IN services), IN services at the periphery (CTI and CSTA based services), IP based Call Centres.

Why? Business & Technical Drivers

- New Service Delivery Model:

A change in the edge of service delivery



examples are switch based CTI extensions, TAPI, PBX based services.

History, Phase 1

- Parlay Consortium initiated in March 98. Originally 5 companies, BT, Microsoft, Nortel, Siemens and Ulticom (called DGM&S on that time)

- **Parlay 1.0** was completed and published in December 98. Parlay 1.0 includes
 - Framework, and the following Service Interfaces

 - Generic Call Control, INAP1 Call Control, Generic Messaging, Generic User Interaction and Call User Interaction

History, Phase 2

- **Parlay 2.0**, six new members were added to start phase 2 in May 99. AT&T, Cegetel, Cisco, Ericsson, IBM and Lucent. Phase 2 completed in Jan 2000
 - Expansion to Phase 1 specifications with
 - Focus on IP & Mobility
 - 2.1 released in Nov '00; Refinement of phase 2 specs. 2.1.1 released in Jan '01
 - Reference Implementations and SDKs (initial discussion)

History, Phase 3

- **Parlay 3.0**, initiated in June '00 (announcement at Supercomm 2000). Currently **62 members**, 24 full and 38 affiliate. Completed late '01
- 2.1 implementation feedback considered
- Parlay Steering Committee's original recommendations for phase 3 (the experiment, as of today March '02, shows that some of them will go beyond phase 3.0):
 - Extending the Parlay API set
 - Alignment with ETSI, 3GPP, JAIN, ...
 - Expanding Parlay members

History, Phase 3 continued

- Original recommendations for phase 3 ...
 - Definition of Parlay deliverables
 - Developer's & Users' Programs

History, Recommendations for Phase 3

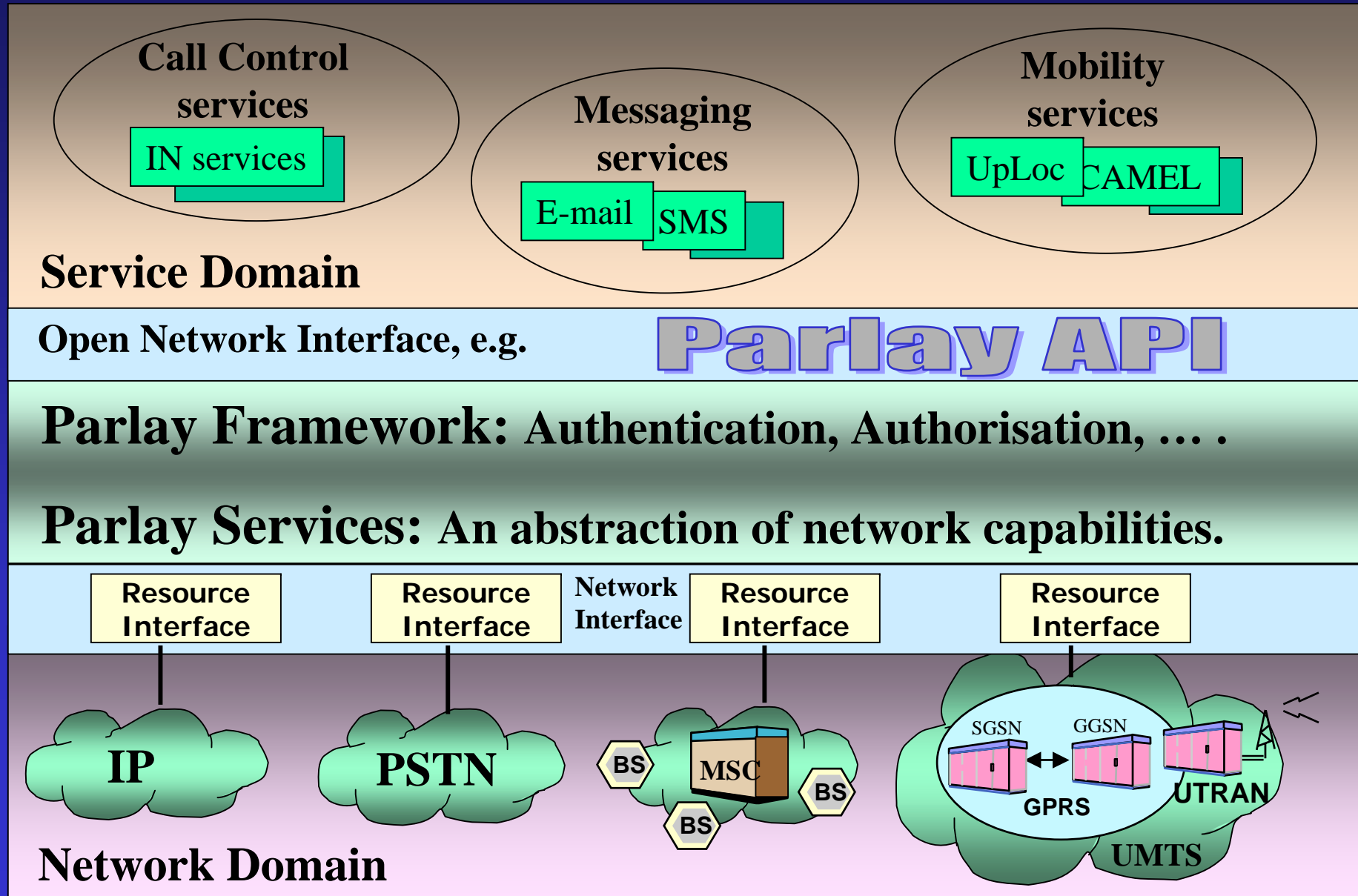
- Extending the Parlay API set
 - New APIs: PAM, Policy Management, Charging, Accounting, Terminal Capability, Data Session Control, M-Commerce
- Alignment with ETSI, 3GPP, JAIN, ...
 - The intent is not to reinvent the wheel ... to use what is already there. There are too many standards ...
- Expanding Parlay members
 - Encourage new members from the telecom and IT industry

History, Recommendations for Phase 3

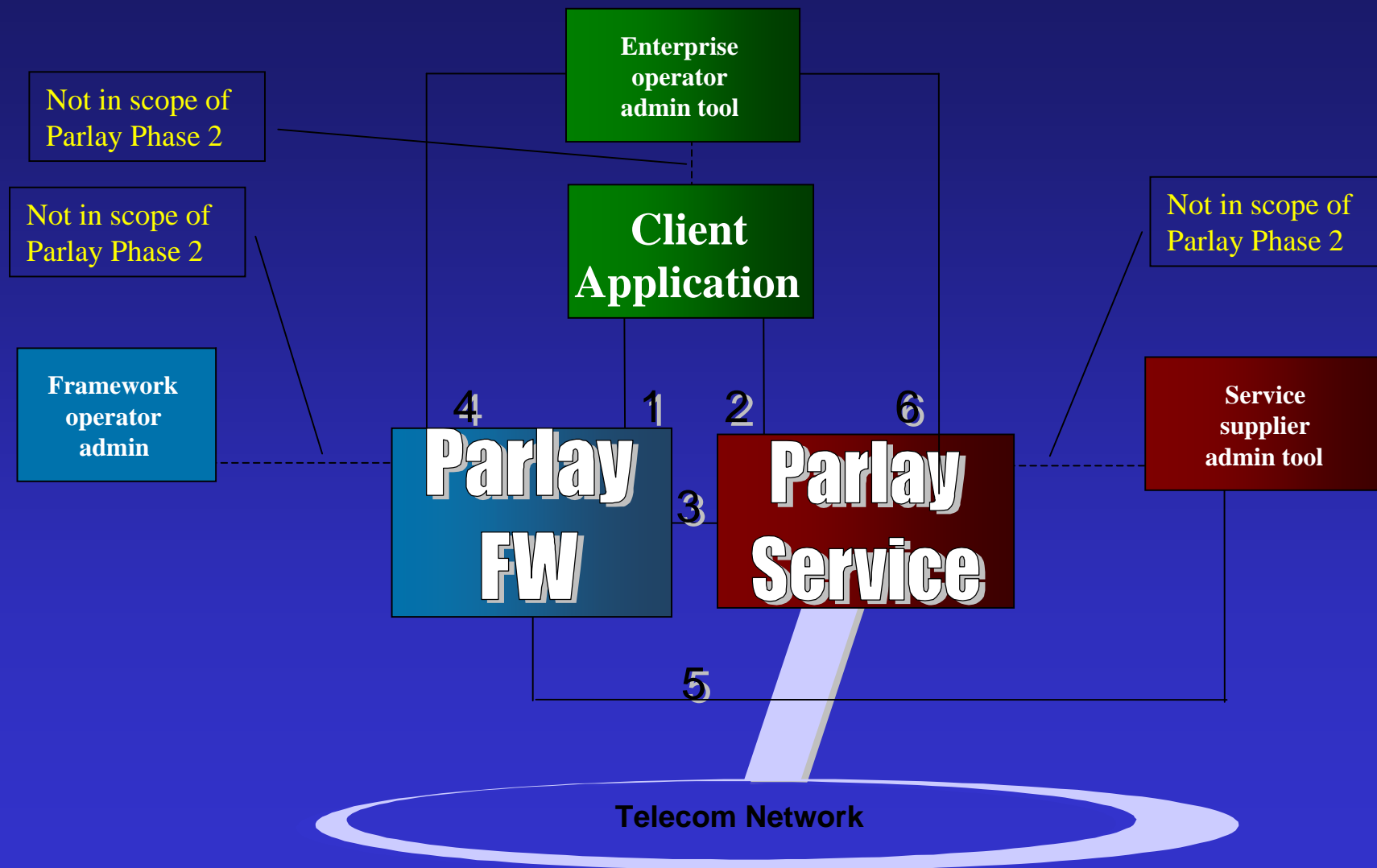
An issue: Parlay APIs are too generic and implementation independent. Problem: interoperability of Parlay products. To address this ...

- **Defining 'Parlay deliverables'**
 - API definitions, SDKs, Reference Implementation and a comprehensive test suite
- **Create a Users' program and a Developers' program**
 - Books, white papers
 - Training materials
 - Conformance & Interoperability test suites
 - Certificate Process

Parlay Architecture



Architecture of Parlay APIs



Source: www.parlay.org

Parlay Architecture - General

- The Parlay APIs are object-oriented and consist of several categories of interfaces as shown in previous slide
- All Applications, Framework and Services Interfaces inherit from the base Parlay Interface Class, 'IparlayInterface'
- Unified Modelling Language, UML, has been used to specify Interface Classes
- The Architecture is based on client/server
- Each Interface consists of a number of Methods. Each Method has a number of Parameters
- Common Data definitions and Interfaces are also defined in OMG IDL

Parlay Architecture - General

- Both synchronous and asynchronous methods are used in APIs:
 - Asynchronous method requests are suffixed by 'req'
 - Asynchronous method responses, if applicable, are suffixed by 'Res' and 'Err'
- The Service and Framework interfaces for client applications are denoted by classes with name `Ip<name>`
- The callback interfaces to the applications are denoted by classes with name `IpApp<name>`

Parlay Architecture - General

- For the interfaces between a Service and the Framework:
 - The Service interfaces are typically denoted by classes with name IpSvc<name>
 - The Framework interfaces are denoted by classes with name IpFw<name>
 - Some methods within Authentication and Access Interfaces are exceptional to this, e.g. IpClientAccess

Parlay Architecture – Two Main Interface Sets

1. Framework Interface Set:

Provides 'surround' capabilities necessary for the Service Interfaces to be open, secure, resilient and manageable

2. Service Interface Set:

Offers applications access to a range of network capabilities

Parlay Architecture – Framework Interfaces

- **Authentication**

Online Authentication of User Application and Network.

- **Authorization**

Access management and Control to Network Services.

- **Discovery**

Capability by which Network Service(s) identity is exposed to a User Application.

- **Event Notification**

Capability by which user application is notified of service related events.

- **Integrity Management**

Capability by which information on events which affect the integrity of the API is shared with the Framework interface and the user application.

Parlay Architecture – Service Interfaces (3.0)

- **Call Control**

Call Management by User Application. Consists of Generic CC, MultiParty CC, MultiMedia CC (SIP enabled CC), and Conference CC.

- **User Interaction**

Management of User Application interaction with Network Services, e.g. Prompt&Collect DTMF, WAP push, etc.

- **Mobility**

Capability to access Mobility information, e.g. Location, Status.

- **Terminal Capability (New in 3.0, adopted from 3GPP OSA)**

Capability to access user's terminal information in the format specified in W3C and adopted in WAP UAProf Specification.

- **Data Session Control (New in 3.0, adopted from 3GPP OSA)**

Management of data sessions in Packet Switching networks, e.g. PDP Context in GPRS.

Parlay Architecture – Service Interfaces (3.0)

- **Generic Messaging**

Capability to send, store, and receive message.

- **Connectivity Management**

Management of IP based connections, including QoS. Partially overlap with Policy Management.

- **Accounting (New in 3.0)**

Capability to get subscriber accounting information for external billing engines.

- **Charging (New in 3.0)**

Capability to update or monitor a balance and generate CDRs for postpaid and prepaid subscribers.

- **Policy Management (New in 3.0)**

Management of static (SLA) and dynamic (per call) policies for network service providers and for 3rd party application service providers.

Parlay Architecture – Service Interfaces (3.0 +, 4.0)

- **Presence & Availability Management, PAM**

Capability of getting presence information, subscriber availability, and also registration of presence reports.

- **Directory/User Profile**

Capability to access subscriber information. In general to access any information held in database.

Parlay & other standardisation bodies

■ Java APIs for Integrated Networks (JAIN)

- JAIN initiated about the same time as Parlay 1.0. The same space. They found each other very quickly. Close cooperation on CC. On the other hand, Parlay is an architecture, which can be filled in by many component based JAIN technologies, like JSLEE

■ ETSI

- A main contributor to telephony standards. At a point, decided to leave out the most 3G standardisation to 3GPP

■ 3GPP

- 3GPP CN5 started to define OSA specs for UMTS R'99 in Nov '99. Due to aggressive time scale, they decided to base their OSA API on the existing industrial Parlay APIs. 3GPP OSA API set is a subset of Parlay API set with some modifications and additions (e.g. Terminal Capability)

Parlay & other standardisation bodies

■ **OMG**

- Its contents strongly influenced by TINA-C and Parlay
- Parlay Framework 2.0 influenced by OMG Telecom Service Access & Subscription, TSAS.
- Parlay Interfaces are defined in OMG IDL
- OMG & Parlay have worked together to keep two standards in synch

Since Oct 00, Parlay CC WG, ETSI SPAN12 (OSA) and 3GPP CN5 (OSA) have formed a Parlay/ETSI/3GPP joint WG. This WG was expanded to cover FW and some other areas. As a result, Parlay 3.0, 3GPP OSA R'4 and ETSI OSA 201-915 have a big common denominator (i.e. the same OSA API set).

Parlay Service Examples

- **Please refer to my presentation titled ‘Proof-of-Concept Parlay Services’ from my home page.**

(Note: These examples are Parlay 2.1 services, not Parlay 3.0)

Are there any concerns, yes there are ...

- phase 1, 2 and 3 are not backward compatible. Has been mandated that the subsequent releases must be backward compatible
- In Author's opinion, phase 3.0 is the first version the industry can rely on. Phase 1 was only a proof-of-concept, phase 2 was a prototype subject for major Parlay players (mainly a number of Parlay members)
- Data types and some interfaces like Call Control are complex
- Lack of implementation experience, especially on performance and dimensioning
- Still concerns and comments about security and integrity

Links ...

www.parlay.org

www.3gpp.org

www.omg.org

www.etsi.org

www.sun.com (search for JAIN)