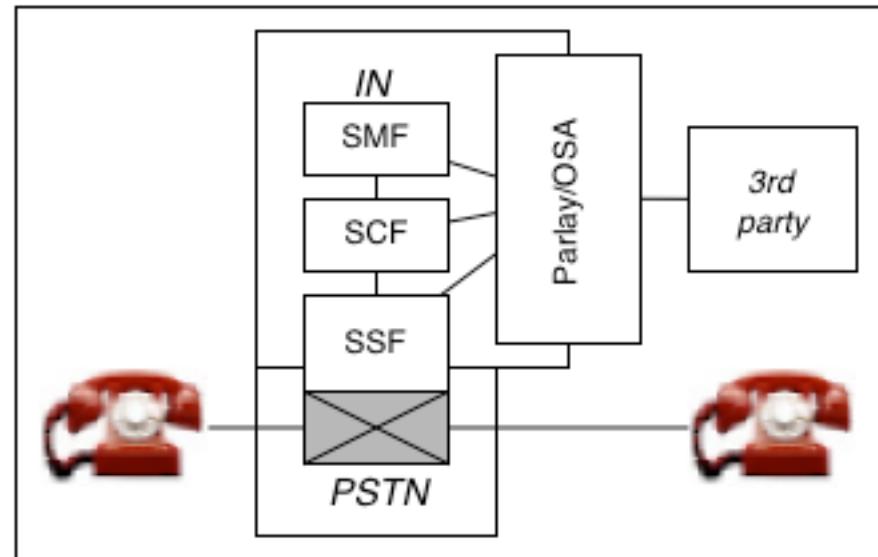
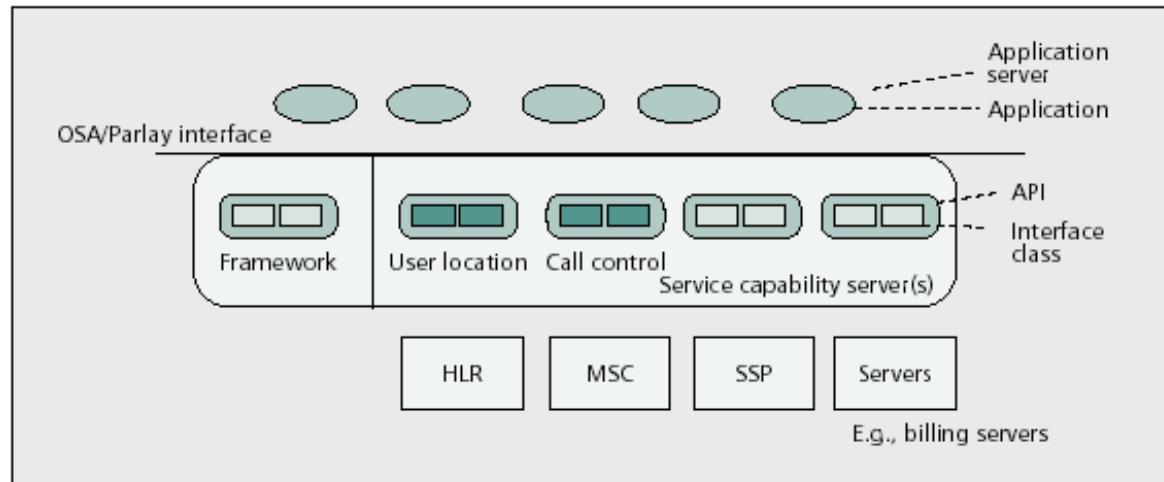


Abertura dos serviços: Parlay/OSA

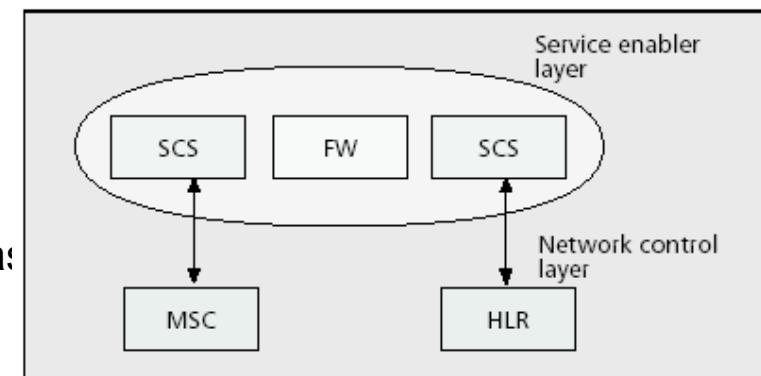
- **Ideia chave:** abrir o desenvolvimento de aplicações de telecomunicações
 - A outras entidades que não os operadores ou fabricantes (*3rd party*)
 - Definição de APIs normalizadas para o acesso a rede e criação de serviços
 - a um maior leque de pessoas...
 - Desenvolvimento de aplicações sem o conhecimento aprofundado de telecomunicações (abstracção dos protocolos e tecnologias de telecomunicações)
- Parlay/OSA criado pelo Parlay Group: 3GPP e ETSI
 - Aplicação: Serviço usado pelo utilizador final (*end-user*)
 - API: Interface, em termos de IN corresponde ao acesso ao SCF
 - Segurança é um aspecto fundamental (acesso total vs. apenas notificações)



Entidades lógicas Parlay/OSA

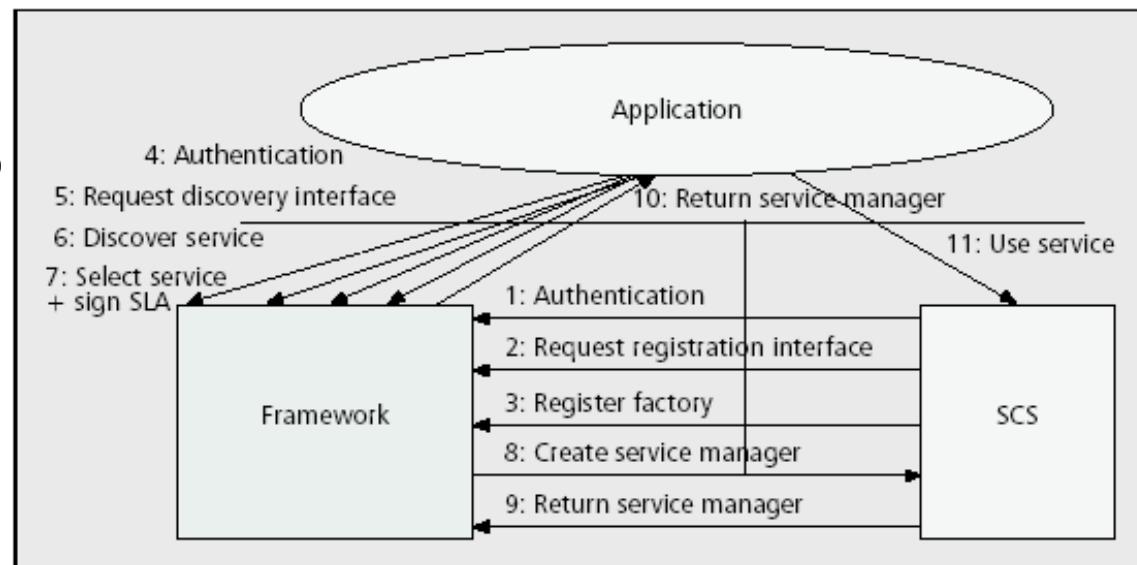


- SCS implementam a parte servidor e as Aplicações a parte cliente das APIs
- SCS interagem com os elementos chave da rede (e.g., SSP, HLR)
- A *Framework* controla e medeia o acesso das aplicações aos serviços
- OSA acrescenta ao Parlay recomendações para o mapeamento das interfaces nos protocolos de acesso à rede



A API Framework

- Inspirado na arquitectura TINA
- Acesso controlado as APIs
 - Autenticação e segurança
 - Descoberta de aplicações
 - Inclusão de múltiplos vendedores e APIs adicionais não normalizadas
 - Núcleo central de interfaces:
 - *Trust and security management*: autenticação de domínios
 - *Registration*: registo de novos SCSS
 - *Service life cycle management* e *Service discovery* criação de novas API e descoberta de API
 - *Contract management*: gestão de contratos entre os diferentes participantes
- 1-3: Registo do SCS com a *Framework*
- 4: Autenticação do domínio da aplicação e verificação do contrato de serviço
- 5-6: Descoberta do serviço
- 7: Estabelecimento do contrato do tipo SLA
- 8-9: Criação de uma instância do serviço
- 10-11: Utilização do serviço



As aplicações Parlay/OSA

- Servidores de aplicações
 - Suporte para a abstração da distribuição (ex., CORBA)
 - Mapeamento das interfaces no protocolo de acesso aos elementos da rede (normalizado no caso do OSA)
 - Suporte para API em linguagens de programação comuns (ex., JAIN, JavaBeans)
 - Disponibilização de SDK (ex., Parlay/OSA SDK da Ericsson)

As interfaces Parlay/OSA

API	Descrição
Framework	Segurança, integridade, autenticação
Call Control	Estabelecimento, fecho e gestão de chamadas, conferencias e chamadas multimédia. Notificações associadas a chamadas ou ligações. Diferentes especializações (Generic Call Control, Multiparty Call Control)
Data Session Control	Estabelecimento, fecho e gestão de sessões de dados (ex., transferência de ficheiros)
User Interaction	Interacção com o <i>end-user</i> para tocar ou apresentar mensagens e/ou recolher informação introduzida pelo <i>end-user</i>
Mobility	Obtenção de informação sobre a localização e/ou estado do <i>end-user</i>
Generic Messaging	Envio e recepção de mensagens (email, voz, SMS, MMS). Gestão de caixas de correio, voz
Terminal Capabilities	Obtenção de informação sobre as capacidades de um terminal (ex., tipo e resolução do ecrã)
Connectivity Management	Negociação e gestão da qualidade de serviço (QoS) e contratos de nível de serviço (SLA) em redes IP
Account Mgmt.	Criação, remoção e gestão de contas de assinantes
Charging	Reserva ou cobrança de fundos de acordo com a conta de um assinante

As interfaces Parlay/OSA

- Especificadas e descritas utilizando a orientação aos objectos
 - Normativo
 - Especificação de interfaces (IDL) e definições de tipos de dados (IDL)
 - Informativo
 - Diagramas de classes (UML) e diagramas de sequências de estados (UML)
- Estrutura genérica de uma interface
 - Implementação da interface em duas partes (rede e aplicação)
 - Do lado da rede
 - `Ip<Interface>` operações para controlar os recursos da rede
 - `Ip<Interface>Manager` criação e gestão de instâncias de `Ip<Interface>`.
Pedidos de notificações de eventos no servidor da aplicação
 - Do lado da aplicação (analogia *callback* interfaces)
 - `IpApp<Interface>` recepção resultados e notificações da parte rede da interface
 - `IpApp<Interface>Manager` criação/gestão de instâncias de `IpApp<Interface>`

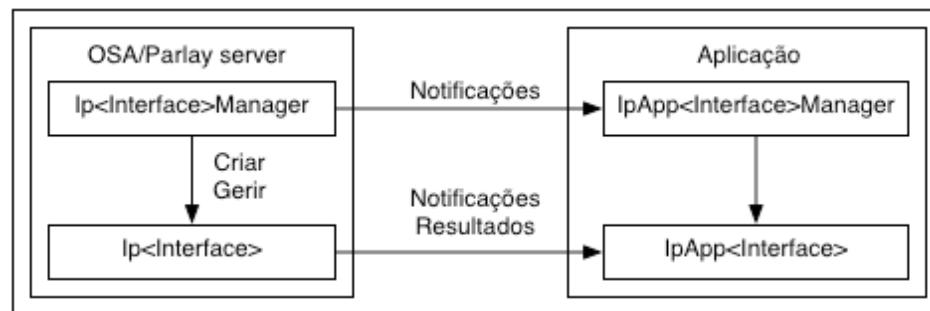


Diagrama de classes da interface *CallControl*

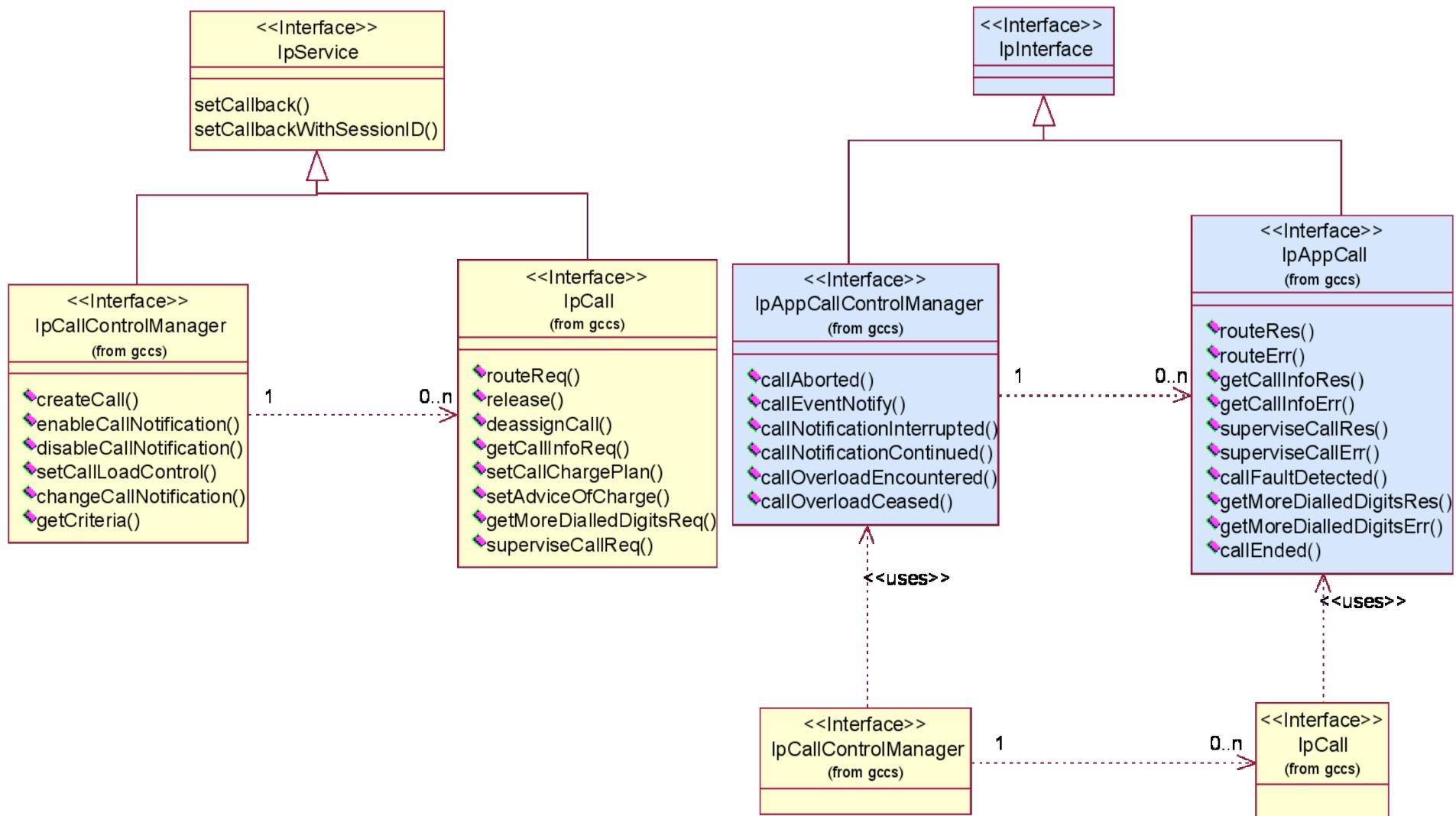


Diagrama de sequência para a interface *CallControl*

- Classe **IpAppLogic**
 - Onde se encontra a lógica do serviço
 - Faz pedidos

Criação e libertação da chamada (2,10)
Estabelecimento de ligações (5,8)

 - Recebe notificações de eventos (6,9)
- Classe **IpAppCall**
 - Recebe notificações/resultados da parte rede da interface (IpCall)
 - Envia notificações/resultados à lógica de serviço
- Classes **IpAppCallControlManager** e **IpCall**
 - Recebe, pedidos da lógica de serviço e da parte aplicação das interfaces
 - Enviam notificações/resultados à parte aplicação das interfaces

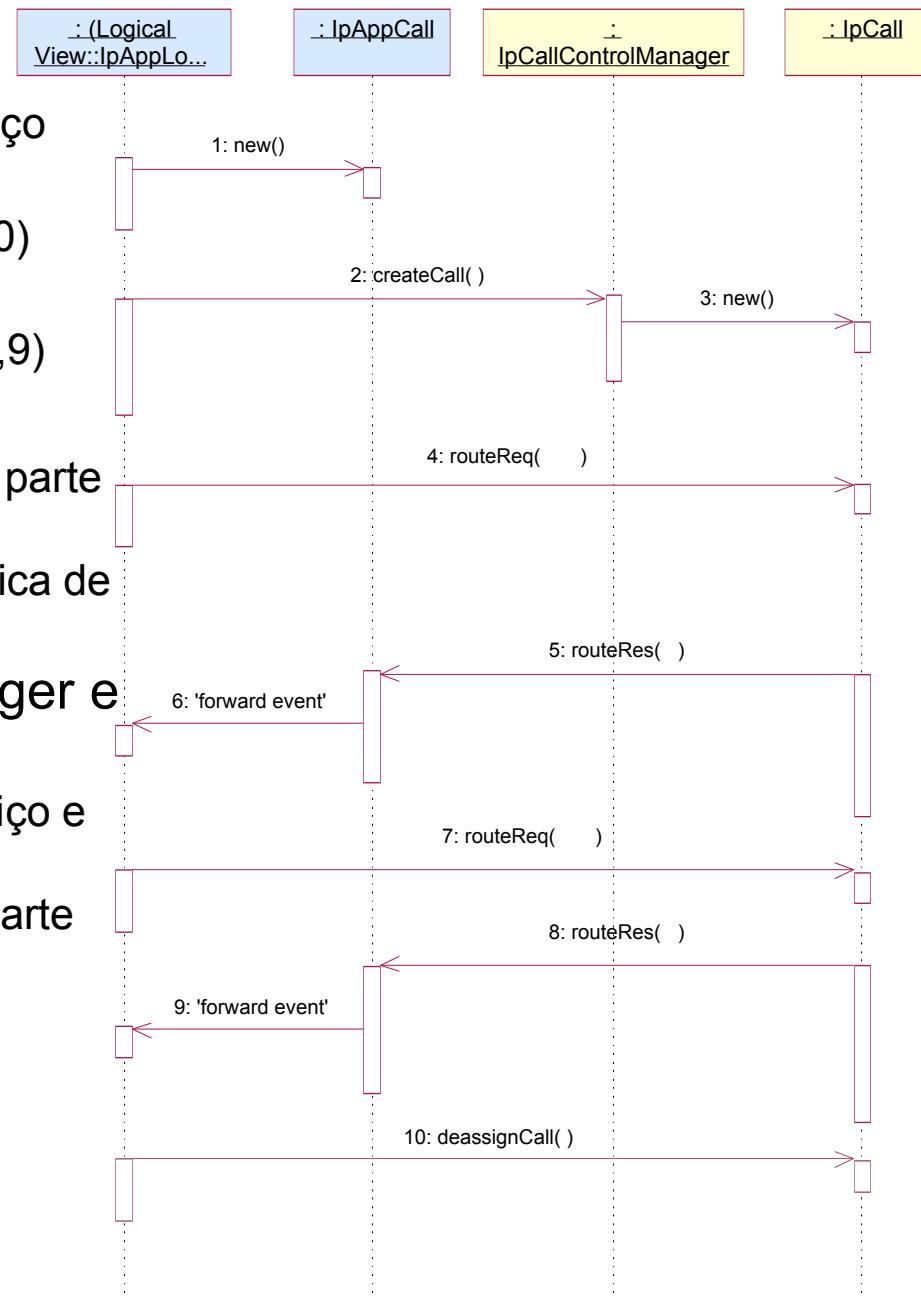
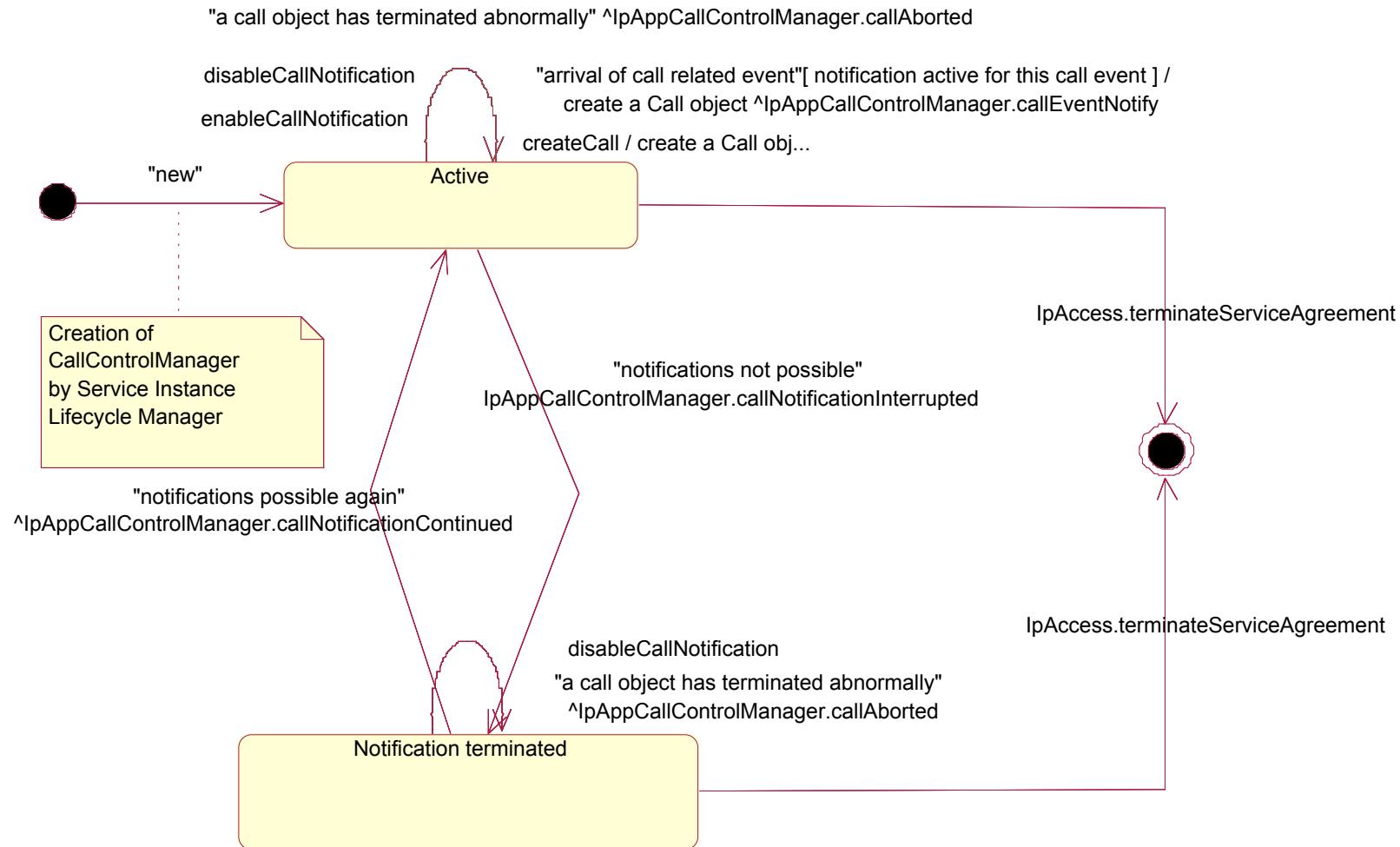


Diagrama estados da classe *IpCallControlManager*



O Network Resource Gateway (NRG) da Ericsson

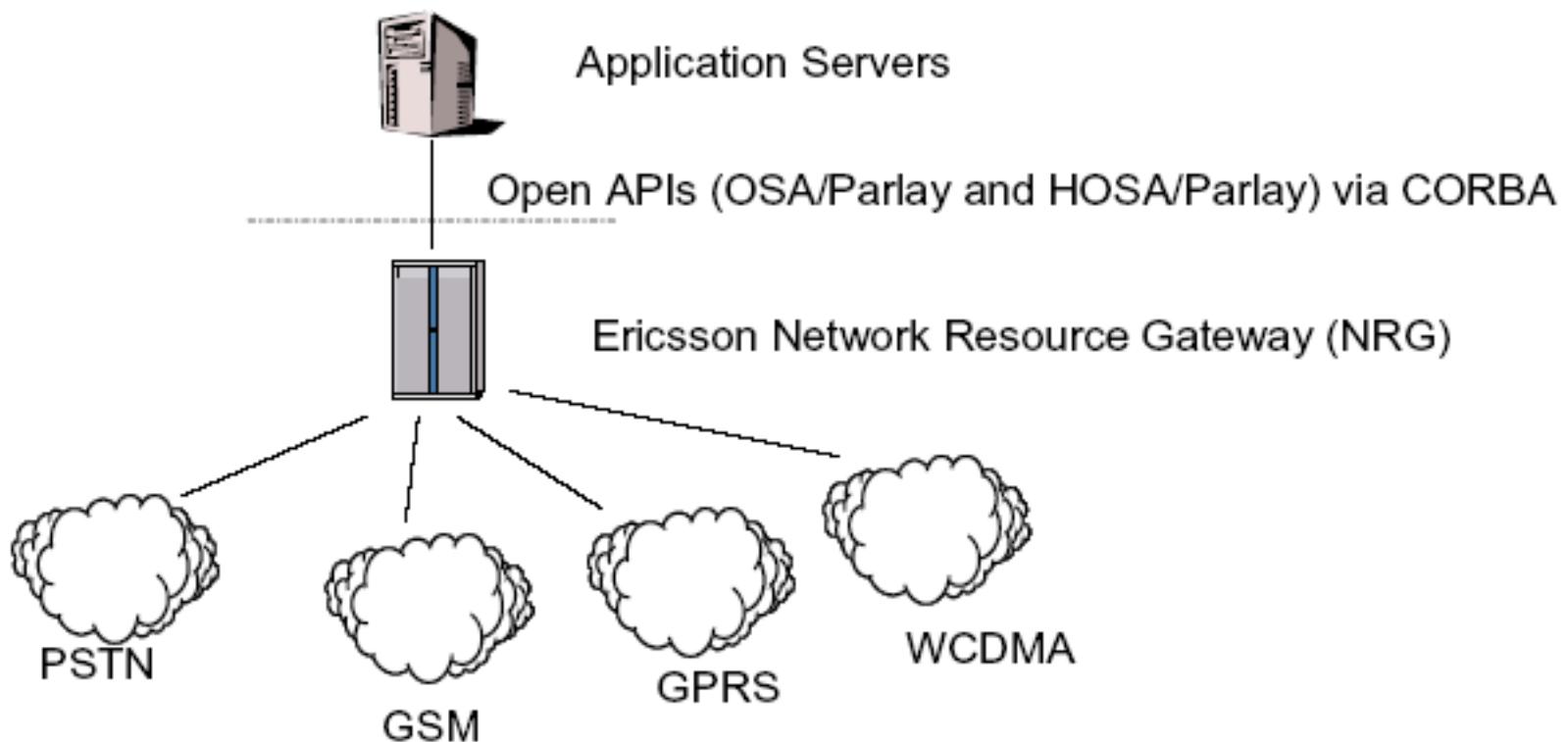
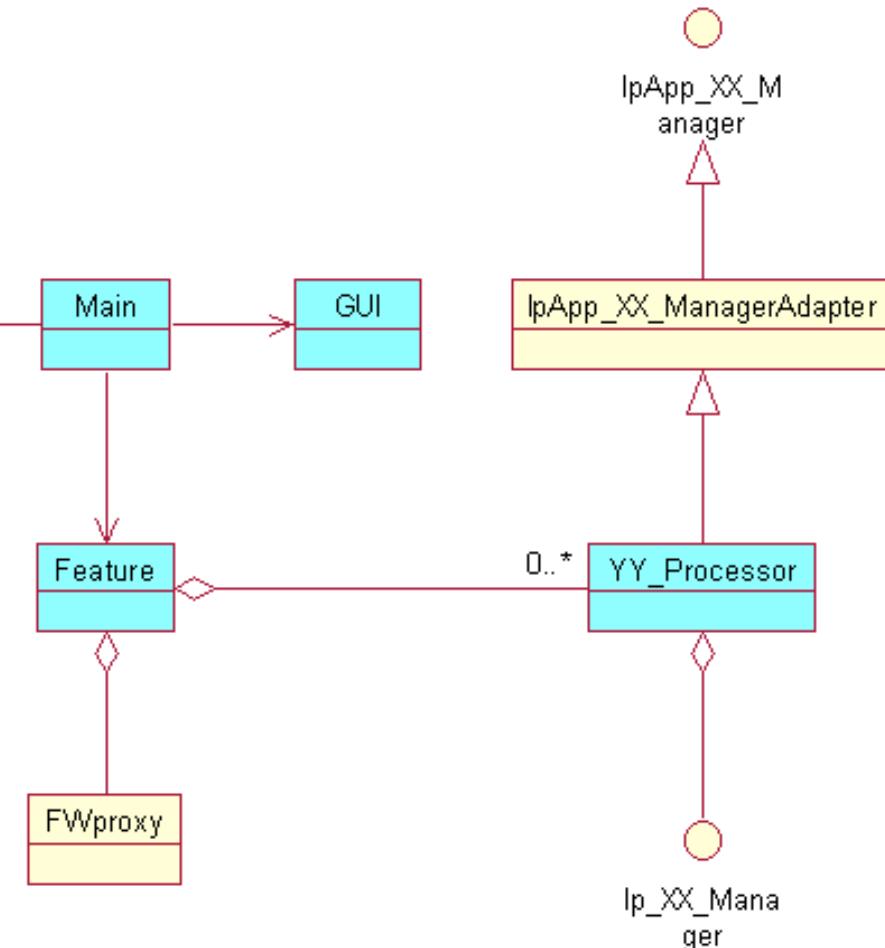


Diagrama de classes das aplicações exemplo do SDK da Ericsson

- Estrutura geral
 - Lógica da aplicação distribuída por várias classes
- Classe Main
 - Responsável por lançar as outras classes (interacção com o FWproxy)
- Classe GUI
 - Interacção com o gestor da aplicação (*start*, *stop*, acções)
- Classe Feature
 - Onde se encontra a lógica do serviço
- Classes YY_Processor
 - Abstraem da classe *Feature* detalhes
 - Pode haver mais de um tipo de YY_Processor numa mesma aplicação
- Classe FWproxy
 - Classe que abstrai alguns dos detalhes da interacção com a *Framework* (ex., autenticação, SLAs)
 - Permite a obtenção e libertação de *service managers*



A classe *Main* das aplicações exemplo

- Classe que abstrai alguns dos detalhes da interacção com a *Framework* (ex., autenticação, SLAs)

- Permite a obtenção e libertação de *service managers*

- Alguns métodos chave

```
// Criação de uma instância de FWproxy,  
// interacção com a Framework  
FWproxy (Properties aProps)  
// Obter um service manager  
public IpService  
obtainSCF (String aSCFName)  
// Libertar recursos  
void releaseSCF (IpService aSCF)  
// Libertar os recursos da Framework  
usados  
public void endAccess ()
```

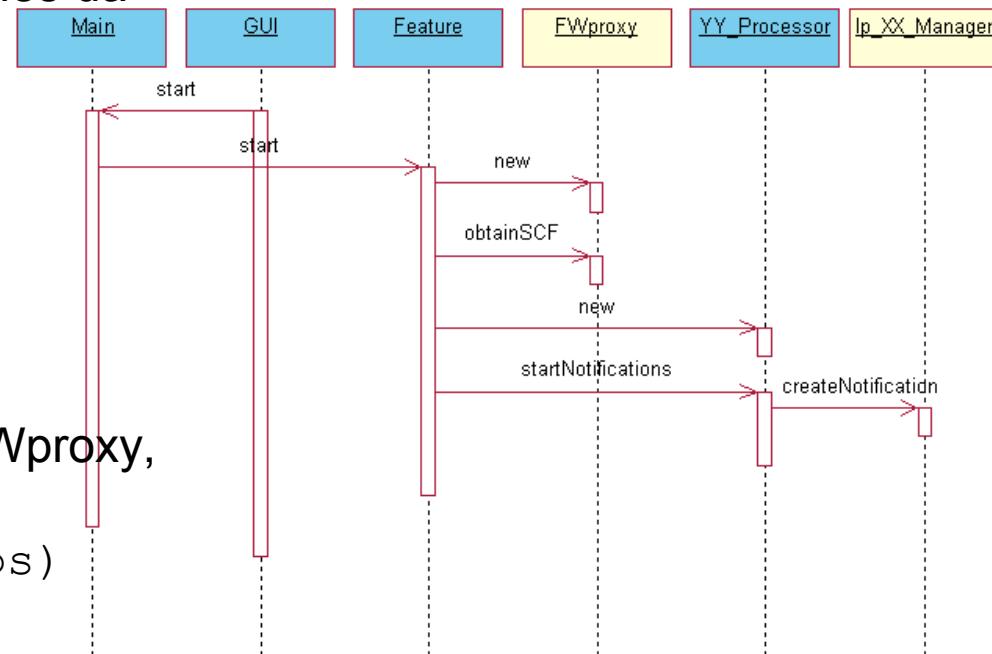
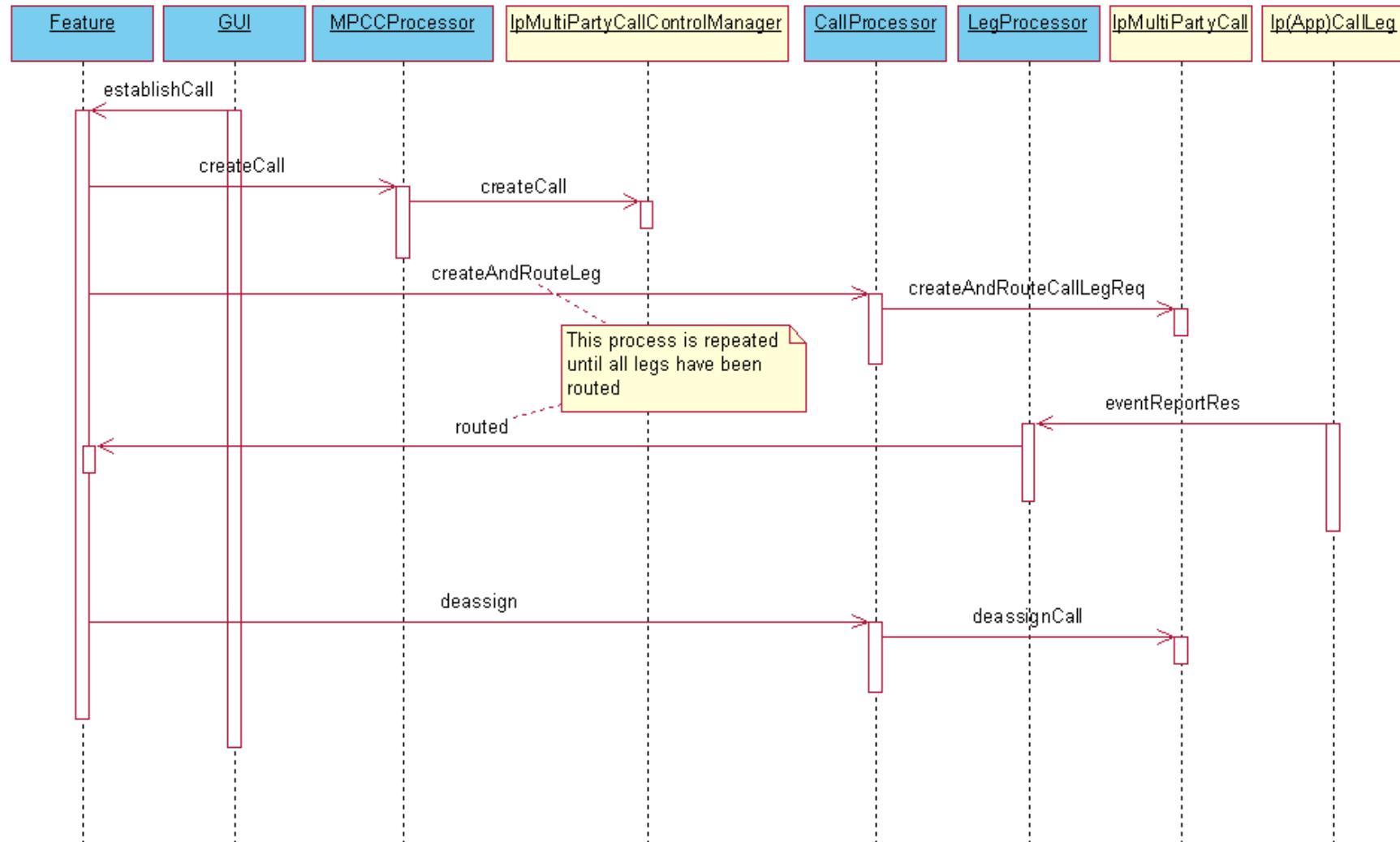


Diagrama de sequência da aplicação exemplo *WebDial* do SDK da Ericsson



Parlay e os Serviços na Web (Web Services): Motivação

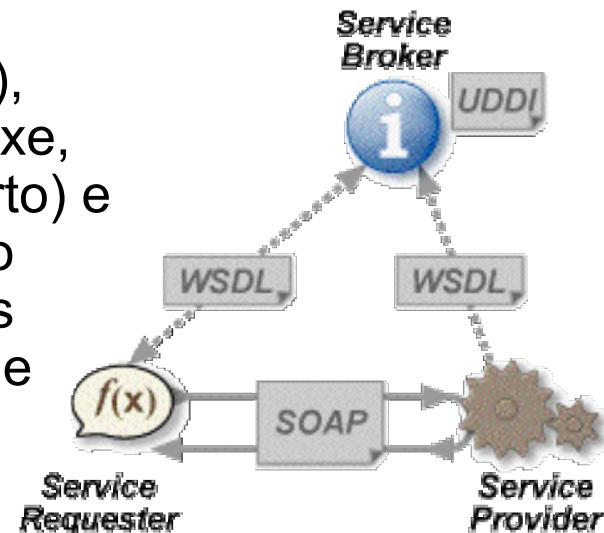
- ***Ideia chave:***

- Utilizar as tecnologias associadas à Web para a criação/utilização de serviços
A escrita de aplicações de telecomunicações tem uma semântica simples e semelhante a outros serviços na Web.
- A interacção entre uma aplicação utilizando um Parlay Web Service e um servidor implementando um Parlay X Web Service é efectuada através da troca de mensagens baseadas em XML.
- Dois tipos de actividades:
 - Parlay Web Services (definição em WSDL das interfaces Parlay)
 - Parlay X (solução totalmente Web, abandono de alguns paradigmas)

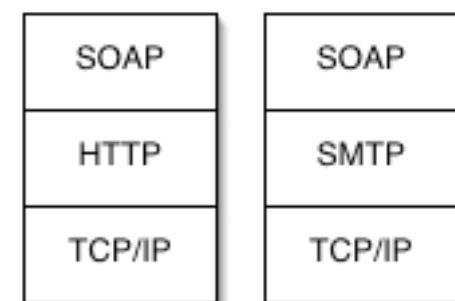
Web Services (tutorial breve): Arquitectura

Elementos da arquitectura:

- **Fornecedores de serviços** (*service providers*), Fornecem um serviços definidos por uma sintaxe, num determinado SAP (ex., endereço IP + porto) e acessível através de um determinado protocolo (ex. SOAP). Estas características são descritas utilizando WSDL e anunciadas a mediadores de serviços.
- **Mediadores de serviços** (*service brokers*) Recebem registo de serviços por parte de fornecedores e pedidos por parte de utilizadores de serviços fazendo o emparelhamento entre ambos.
- Clientes de serviços. Fazem a descoberta dos prestadores dos serviços desejados consultando mediadores de serviços. Após essa descoberta utilizam esses serviços de acordo com a sua descrição fornecida (em WSDL).



Pilha Protocolar SOAP



WSDL: Web Services Description Language (v1.1)

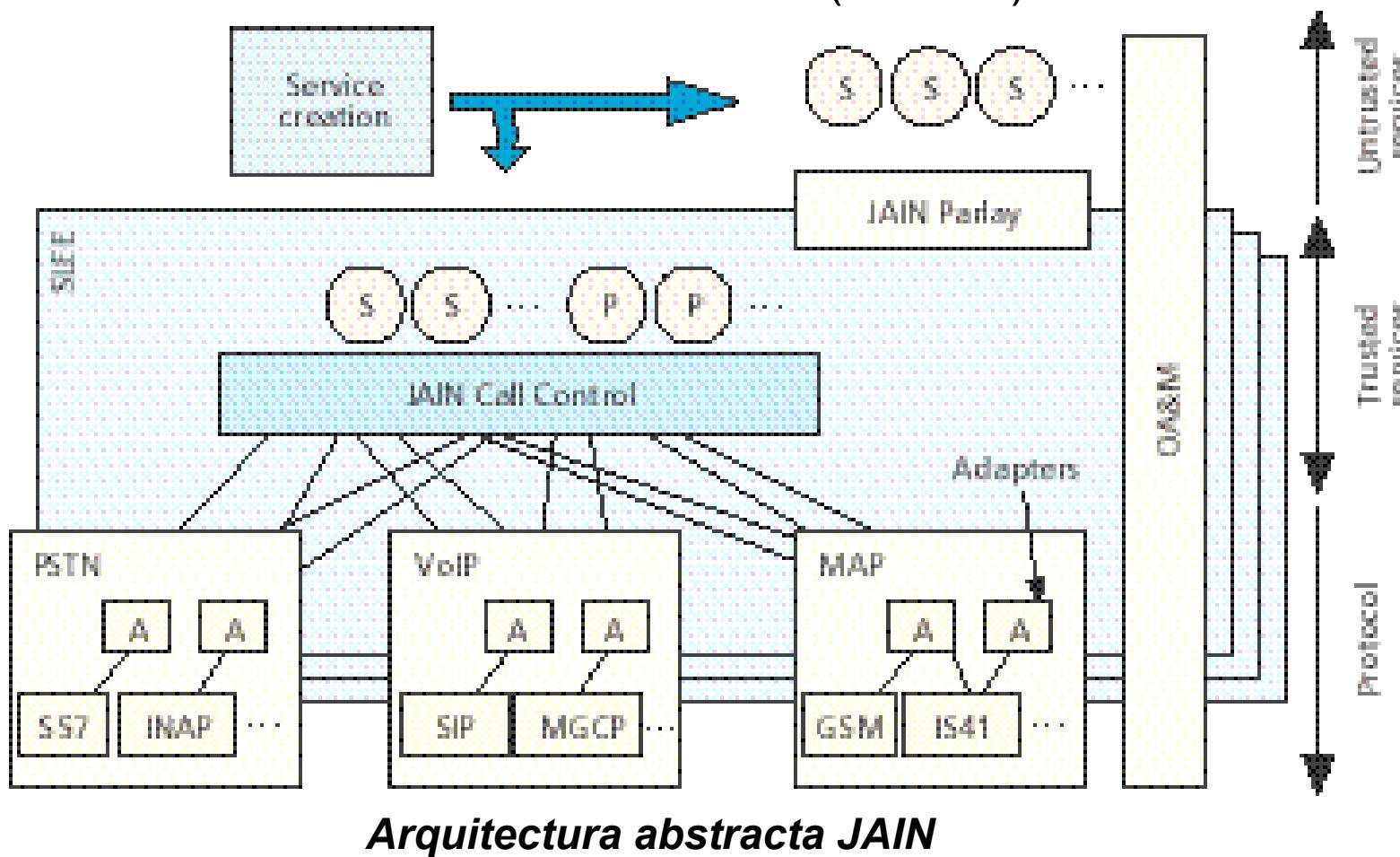
- WSDL define **serviços** através de colecções de portos (**endpoints**).
- Esta definição é independente da sua implementação prática (formato das mensagens e protocolos utilizados).
- Elementos do WSDL (utilização de XML):
 - **Types**: definição de tipos de dados simples e complexos
 - **Messages**: descrição **abstracta** dos dados trocados na prestação de um serviço,
 - Composta por várias **parts** (ex.,..header, body)
 - **Operations**: descrição **abstracta** das acções suportadas por um serviço
 - **Definida pelas mensagens dos tipos**: **Input**, **output**, **fault**
 - **Dependendo do dos tipos de mensagens utilizadas podem ser definidos quatro modelos de comunicação**:
 - **One-way**, **request-response**, **solicit-response**, **notification**
 - **PortType**: colecção de operações suportadas num ou mais serviços
 - **Binding**: descrição do formato de dados e protocolos usados na implementação de um PortType
 - **Port**: um **endpoint** definido como a combinação de um **binding** e um endereço
 - **Service**: colecção de **endpoints** relacionados

Documentos Parlay X (3GPP)

- [**TS 29.199-01**](#) Open Service Access (OSA); Parlay X web services; Part 1: Common .
- [**TS 29.199-02**](#) Open Service Access (OSA); Parlay X web services; Part 2: Third party call .
- [**TS 29.199-03**](#) Open Service Access (OSA); Parlay X web services; Part 3: Call notification .
- [**TS 29.199-04**](#) Open Service Access (OSA); Parlay X web services; Part 4: Short messaging .
- [**TS 29.199-05**](#) Open Service Access (OSA); Parlay X web services; Part 5: Multimedia messaging.
- [**TS 29.199-06**](#) Open Service Access (OSA); Parlay X web services; Part 6: Payment .
- [**TS 29.199-07**](#) Open Service Access (OSA); Parlay X web services; Part 7: Account management.
- [**TS 29.199-08**](#) Open Service Access (OSA); Parlay X web services; Part 8: Terminal status .
- [**TS 29.199-09**](#) Open Service Access (OSA); Parlay X web services; Part 9: Terminal location .
- [**TS 29.199-10**](#) Open Service Access (OSA); Parlay X web services; Part 10: Call handling .
- [**TS 29.199-11**](#) Open Service Access (OSA); Parlay X web services; Part 11: Audio call .
- [**TS 29.199-12**](#) Open Service Access (OSA); Parlay X web services; Part 12: Multimedia conference .
- [**TS 29.199-13**](#) Open Service Access (OSA); Parlay X web services; Part 13: Address list management .
- [**TS 29.199-14**](#) Open Service Access (OSA); Parlay X web services; Part 14: Presence .
- [**TS 29.199-15**](#) Open Service Access (OSA); Parlay X web services; Part 15: Message Broadcast
- [**TS 29.199-16**](#) Open Service Access (OSA); Parlay X web services; Part 16: Geocoding and Mapping .
- [**TS 29.199-17**](#) Open Service Access (OSA); Parlay X web services; Part 17: Application driven Quality of Service (QoS)

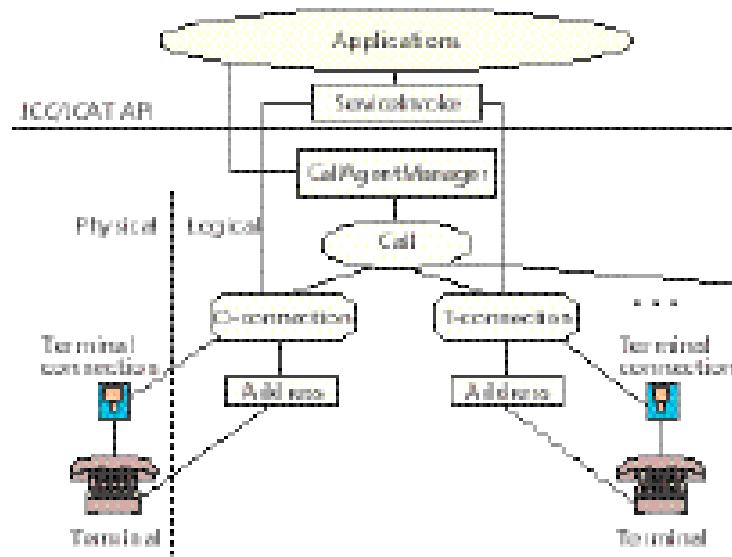
JAIN: Java API for Integrated Networks

- Conjunto de **API** Java aberta, com acesso a “funções” de rede por forma a permitir o desenvolvimento de serviços, potenciando a sua portabilidade, convergência e segurança.
- Assume núcleo da rede como sendo IP (ou ATM)



Call Control (JCC)/Coordination and Transactions (JCAT)

- Modelo para as chamadas (*call model*) em que a “rede” é vista como uma máquina virtual.
 - Uma chamada é vista como uma sessão.
 - Combina as características da IN (FSM para as chamadas) e de outras visões das chamadas telefónicas baseadas na sua manipulação como objectos (ex. JTAPI).



Aplicação JAIN (JCC/JCAT) a serviço sobre telefonia

JAIN no contexto das redes e serviços de comunicações

- **Java Service Logic Execution Environment (JSLEE):** núcleo da prestação de serviços permitindo que aplicações ou *Service Building Blocks (SBB)* acedam e manipulem recursos da rede.
- **Resource Adaptors (RA)**
- **Service Building Blocks (SBB)**
- Definição de APIs para a gestão da rede e serviços

