Advanced Conferencing (Multiparty multimedia sessions)

Advanced Conferencing
-- Part I --

Dr F. Belqasmi, PhD. Concordia University
Dr C. Fu, PhD. Ericsson Canada

Part I: Introduction, signalling and media control protocols

- Introduction
  - What is multiparty multimedia session
  - Technical components
- Signaling protocols
  - SIP
- Media control protocols
  - SIP based protocols

- Introduction
  - What is multiparty multimedia session
  - How to implement
  - Protocols involved
  - Classifications

Multiparty multimedia session

- The conversational exchange of multimedia content between several parties
  - About multimedia
    - Audio, video, data, messaging



  - About participants
    - Any one who wants to participates the conference


How – thinking from a real life case

- When organizing a conference or a meeting, what to do?

Deciding topics, participants, time, agenda and booking a conf room → Policy control

Inviting participants and getting their confirmation to attend → Signaling

Starting the conference: let people seat down in the room and prepare the projector, microphone, player → Media control

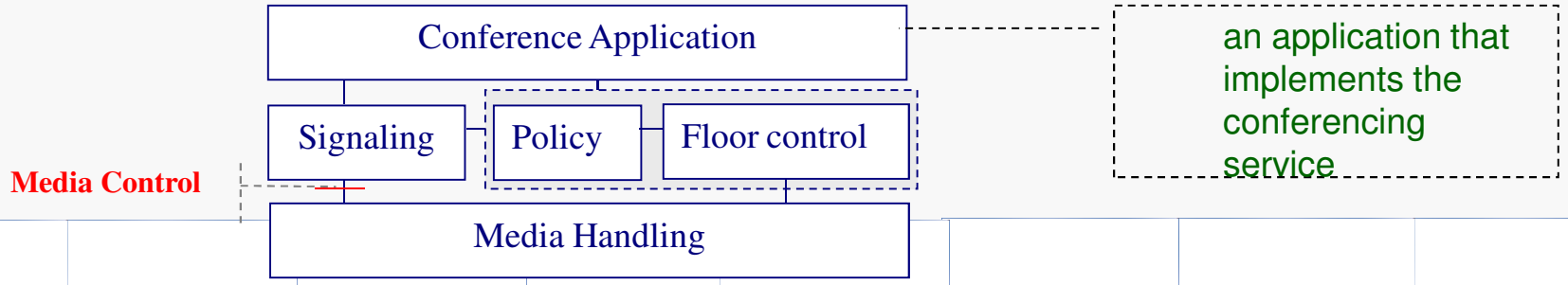
During a conference:

-- talking, discussing; presenting, playing a video to everybody, translating → Media handling

-- being a chair and deciding who can talk next → Floor control

How – technical components

- Signaling
  - Session establishment, modification and termination
  - Capability negotiation
- Media
  - Media handling: media transmission, mixing, trans-coding
  - Media control: stands when there is a separation of signaling and media mixing entities
- Conference control
  - Conference policy: conference arrangement, admission control, participant management, voting
  - Floor control: allows users of share resources such as video and audio without access conflicts.



Protocols involved

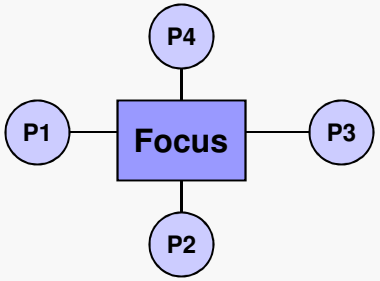
- Signaling
  - H.323, SIP (Session Initiation Protocol)
- Media
  - Media control: Megaco (Media Gateway Control protocol), SIP based media control – NetAnn/SIP MSCML (Media Server Control Markup Language), SIP media control channel framework
  - Media transport: RTP/RTCP, SRTP
- Conference control
  - Policy control: CPCP (conference policy control protocol), XCAP
  - Floor control: BFCP (Binary Floor Control Protocol), TBCP (Talk Burst Control Protocol)
    - Floor server control: FSCML (Floor Server Control Markup Language)

Classifications

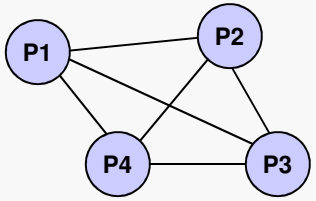
- Open/close
- Pre-arranged/ad hoc
- With/without sub-conferencing (i.e. sidebar)
- With/without floor control
- Topology: centralized, distributed, hybrid

- Signaling protocols
  - IETF: SIP
    - Conferencing models
    - Scenarios

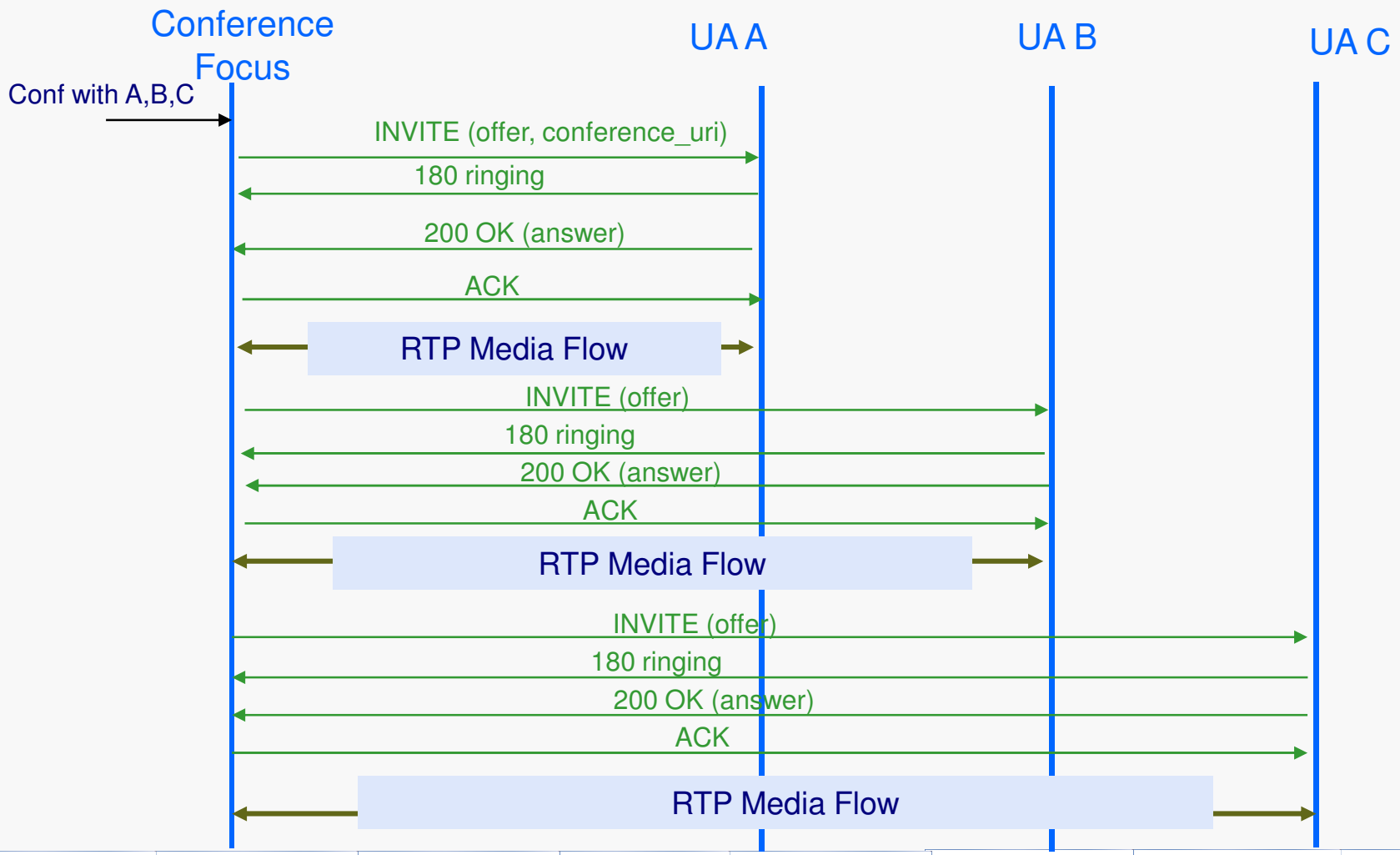
SIP conferencing models



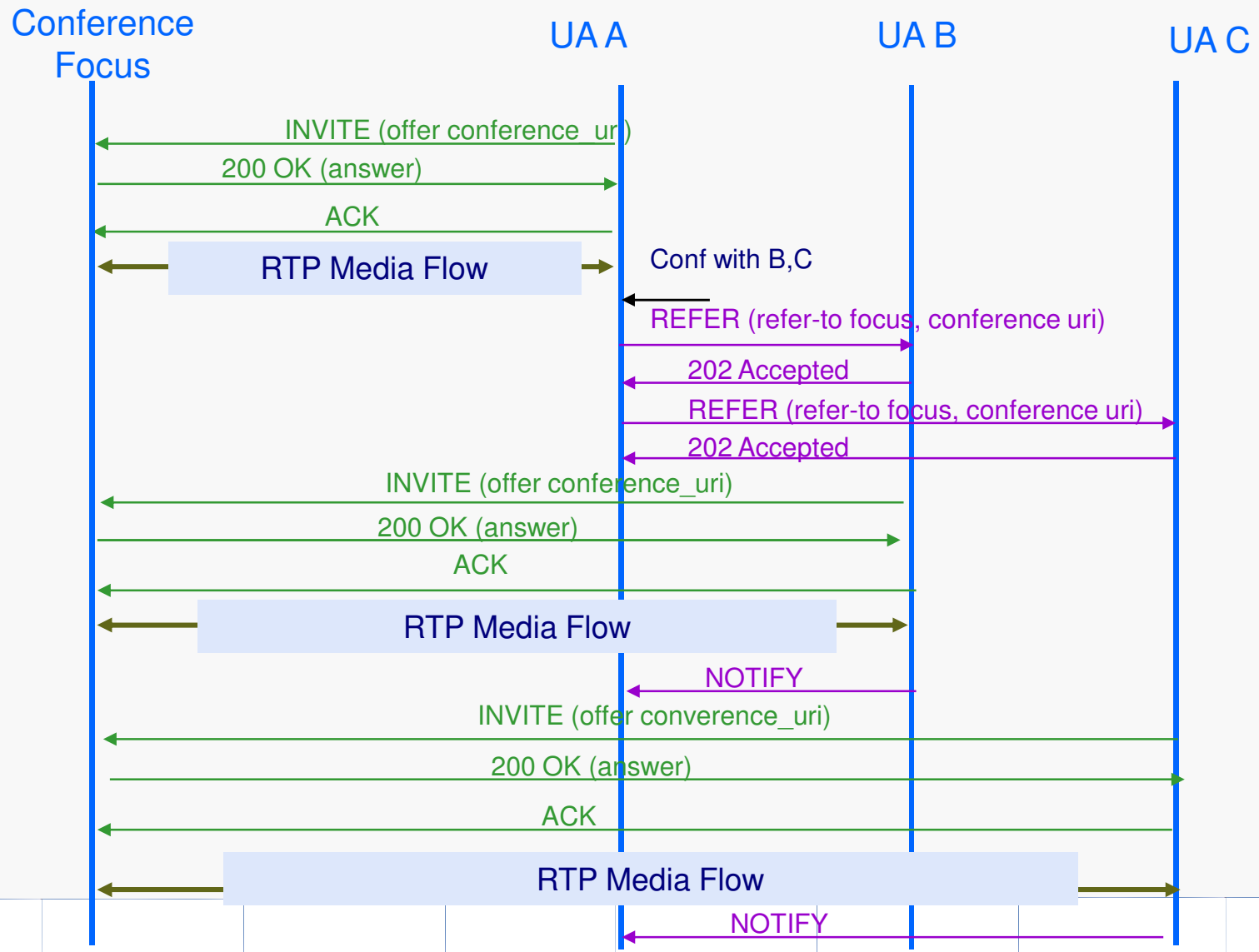
- Tightly coupled conference
  - Dial-In Conference
    - End point invite conference server which handle the media mixing
  - Dial-Out Conference
    - Server invite all the parties into a conference
  - Ad-hoc Centralized Conference
    - Two party setup conference directly, other party added through a conference server
- Loosely coupled
  - central signaling with multicast media
- Fully distributed



SIP conference example – dial out



SIP conference example – dial in



- Media control protocols
  - SIP Based Media Control
    - MSCML
    - SIP media control channel framework

- SIP based media control protocols
  - MSCML (RFC 5022)

What is MSCML

- Defined initially by RFC 4722, replaced by RFC 5022
- provides services to users at an application level, services specified in user part of SIP Request URI, control between AS and MS
- Provide IVR and advanced conference service, as well as fax
- Command oriented, request/response protocol

Basic concept

- There are three type of MSCML message, request, response, notification

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      ... request body ...
    </request>
  </MediaServerControl>
```

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <response>
      ... response body ...
    </response>
  </MediaServerControl>
```

- MSCML messages are located in the body of SIP Request messages. Each SIP request can only embed on MSCML message
  - SIP request messages: INVITE, INFO
  - ‘conf’ and ‘ivr’ in SIP request URI specify the message type

MSCML main commands

- Main requests
  - Conference related
    - <configure_conf>
    - <configure_leg>
    - <configure_team>
  - IVR related
    - <play>
    - <playcollect>
    - <prompt>
    - <playrecord>
    - <stop>
  - Event/signal (within a dialog)
    - <subscribe>
    - <notification>
    - <signal>
- Response
  - ID: optional
  - Request Type: e.g. <play>
  - Code: 2XX, 4XX, 5XX
  - Text: human readable

MSCML conference management

- Configure_conference is mandatory: creating a control leg for conference

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      <configure_conference reservedtalkers="120"
        reserveconfmedia="yes"/>
    </request>
  </MediaServerControl>
```

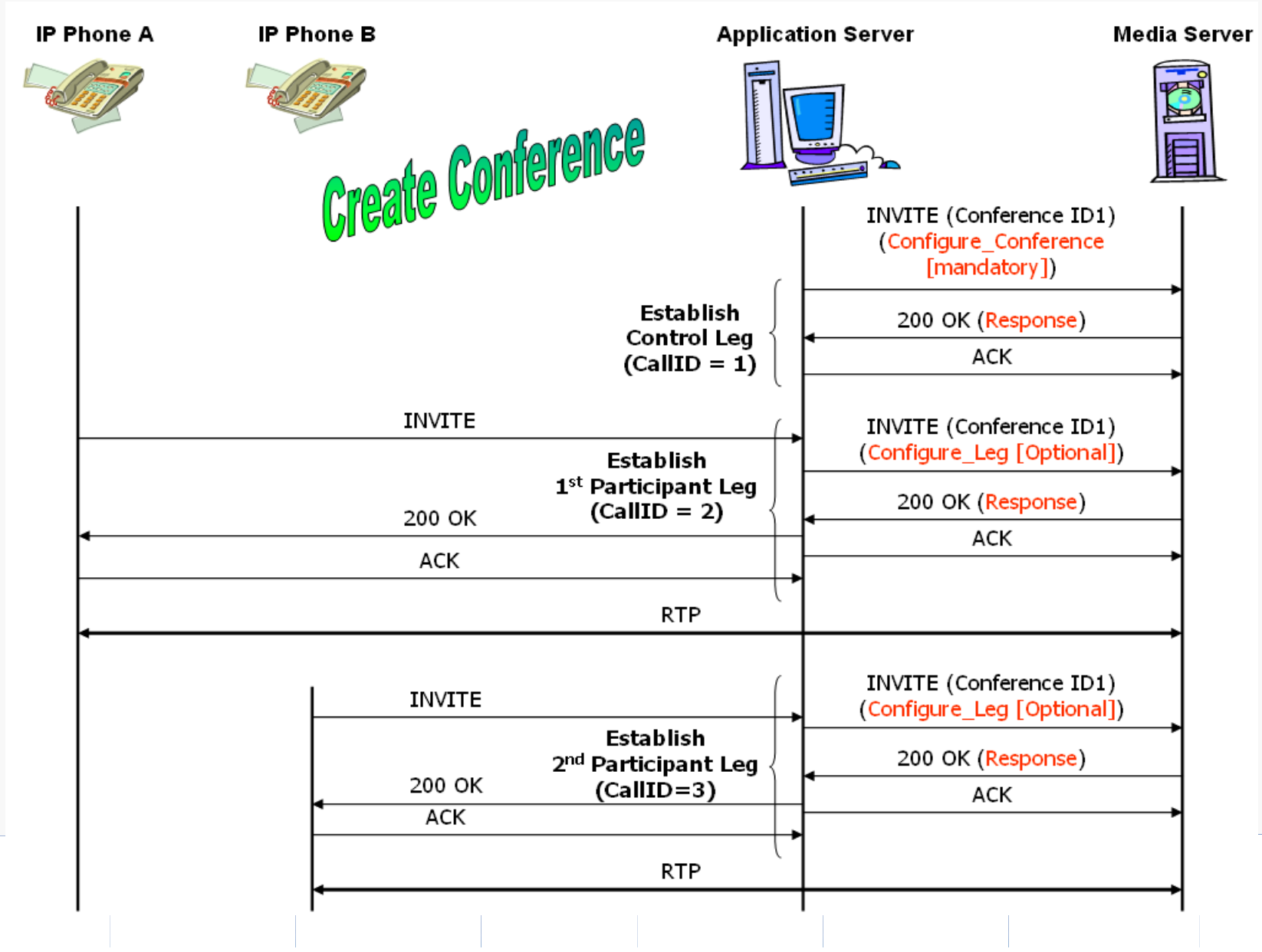
- Configure_leg is a control leg for a dialog. It can configure the dialog's media mode

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      <configure_leg mixmode="mute"/>
    </request>
  </MediaServerControl>
```

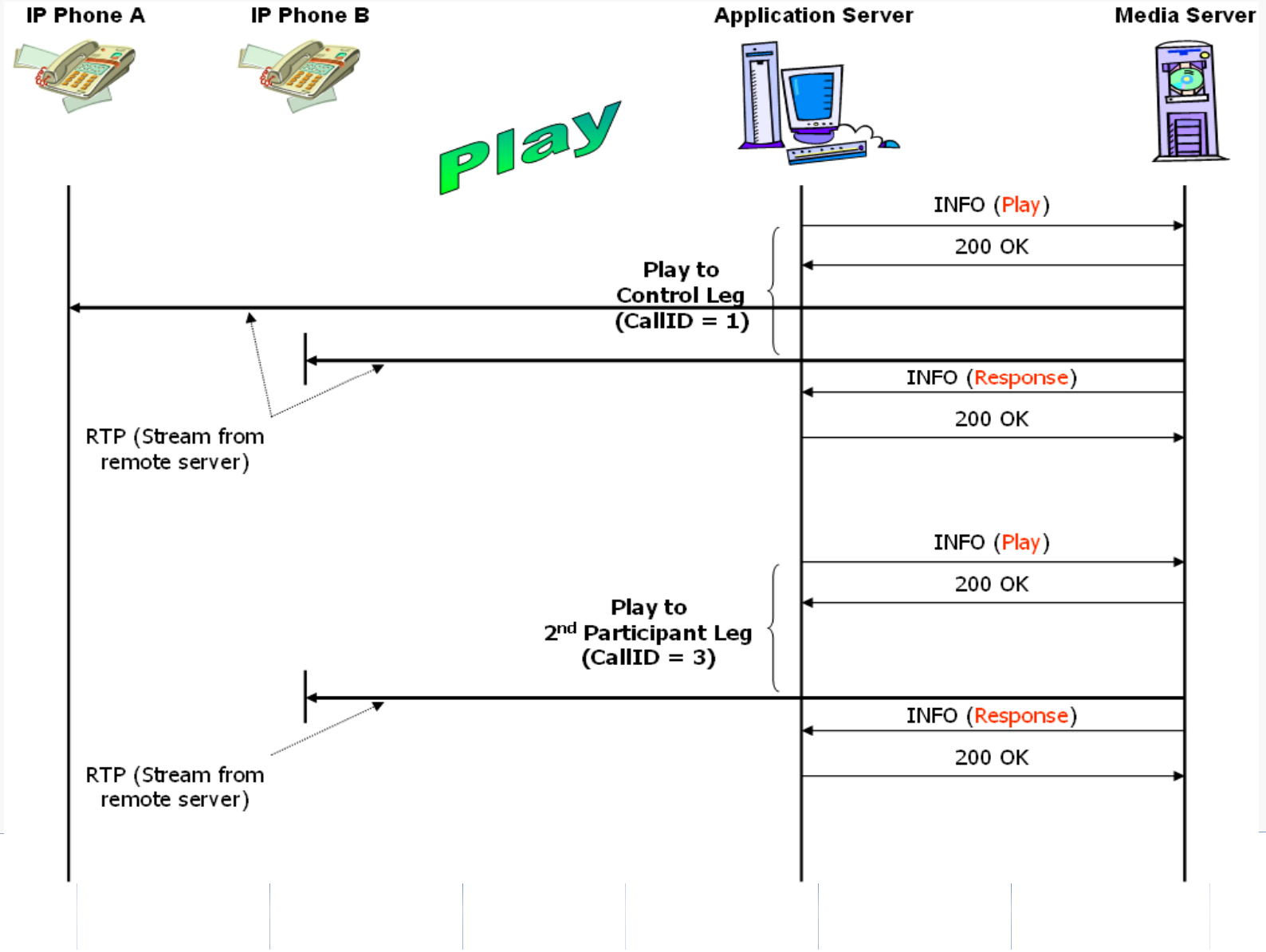
- Can play a prompt to a conference or to a specific leg
- Conference terminates by sending a BYE to conference control leg
- BYE to a leg will just remove a participant

```
<?xml version="1.0" encoding="utf-8"?>
  <MediaServerControl version="1.0">
    <request>
      <play>
        <prompt>
          <audio
url="http://prompts.example.net/en_US/welcome.au"/>
        </prompt>
      </play>
    </request>
  </MediaServerControl>
```

MSCML conference example – Create



MSCML conference example – play a prompt



Questions

Advanced Conferencing
-- Part 2--

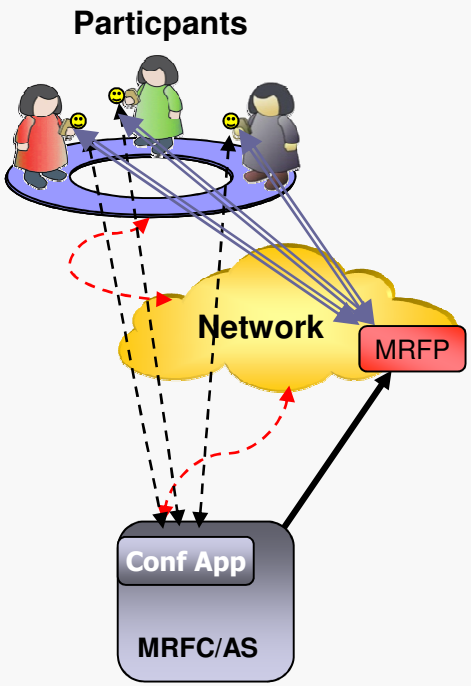
Dr F. Belqasmi, PhD. Concordia University
Dr C. Fu, PhD. Ericsson Canada

Part-I

Multiparty multimedia session

- Is the conversational exchange of multimedia content among several parties.

- It has 3 main building blocks:
  - Signaling
    - H.323, SIP
  - Media control and handling
    - Megaco/H.248, NetAnn/SIP-MSCML
    - RTP
  - Conference control
    - Policy control: CPCP (conference policy control protocol), XCAP
    - Floor control



Agenda

- Floor control
- Putting it together
- Case study

Floor Control

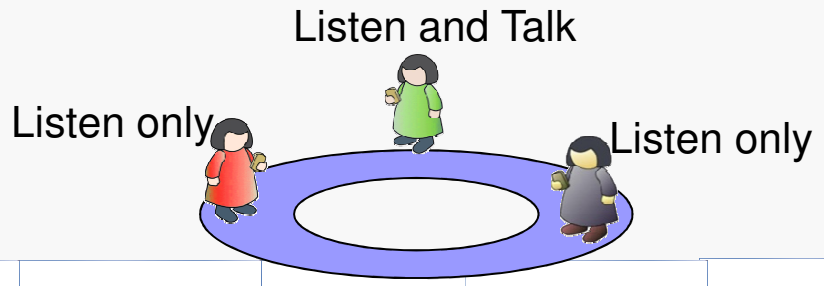
- Definition
- Architecture
- Protocols

Definition

Floor control: a mechanism that enables the management of the joint or exclusive access to the shared resources inside a conference

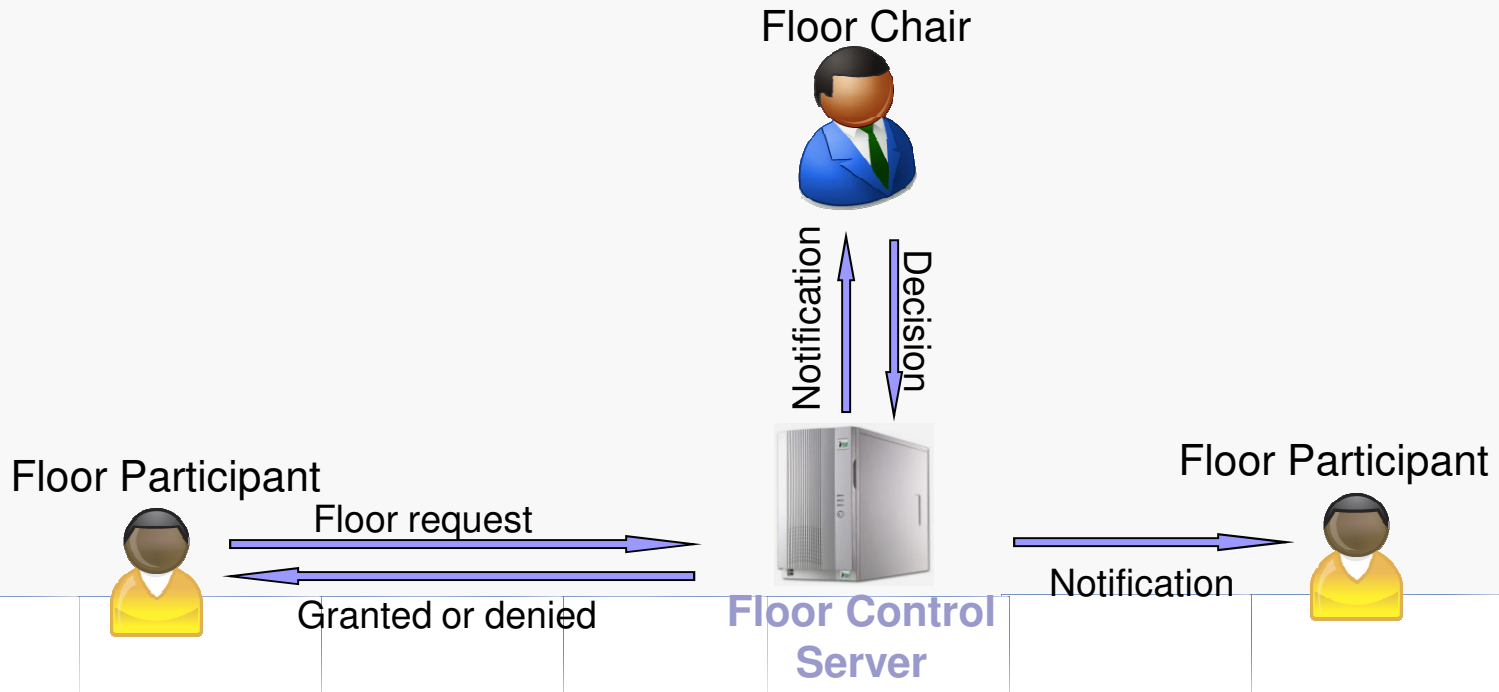
.e.g. audio/video channels, slide bar presentation

Floor: “A temporary permission to access or manipulate a specific shared resource or set of resources”.



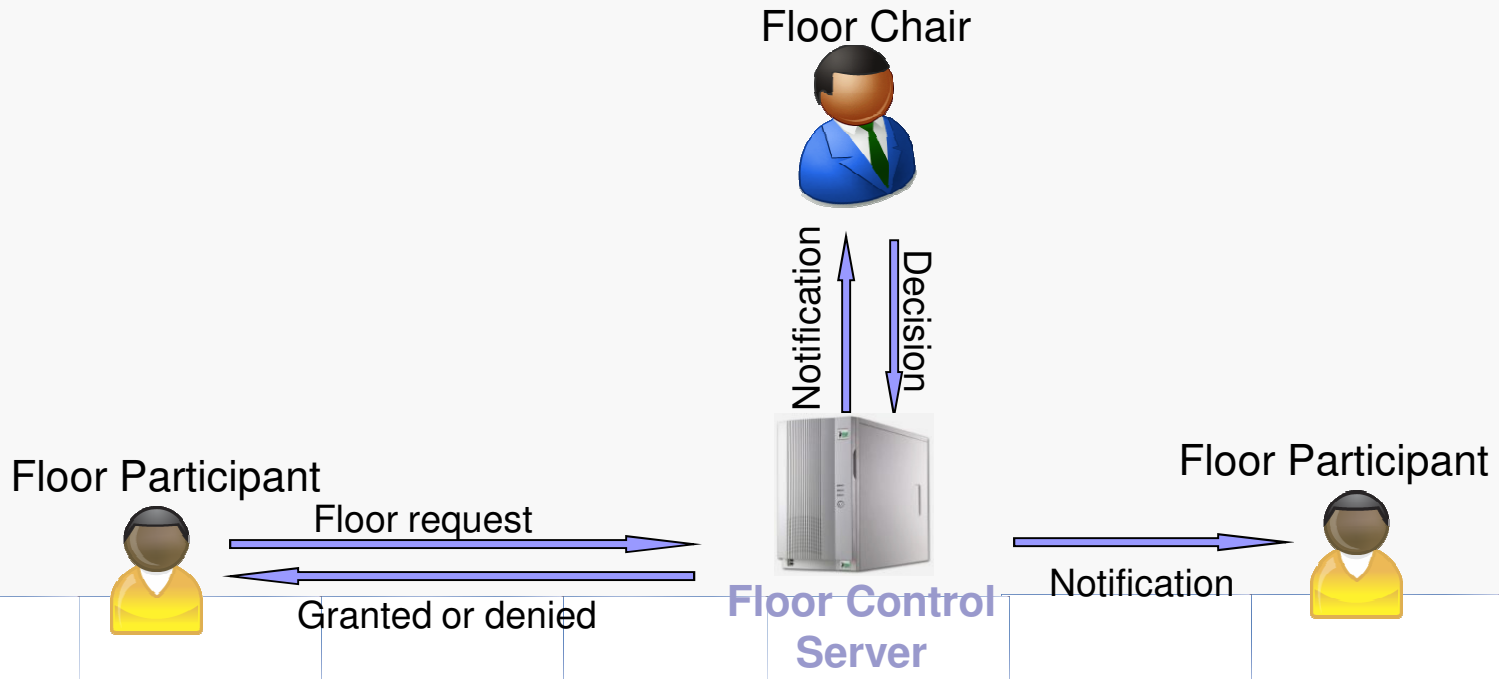
Architecture

- Three entities are involved in floor control:
  - Floor participant
  - Floor chair
  - Floor Control Server (FCS)



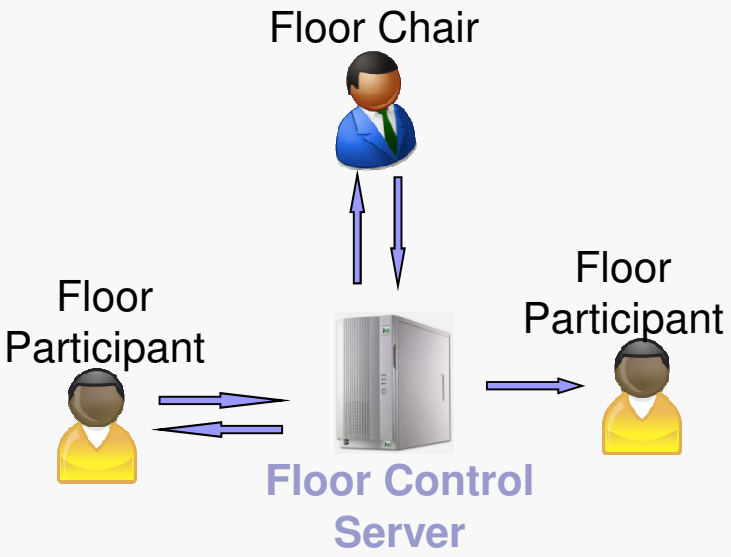
Architecture

- Two main algorithms
  - First come First Serve (FCFS)
  - Chair moderated



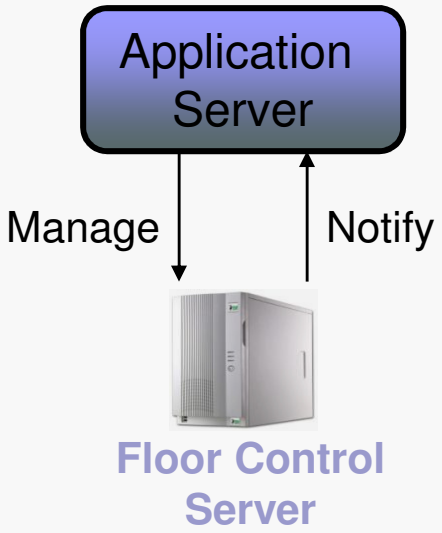
Protocols

1. Establish floor control connections between the different entities



2. Coordinate access to shared resources

3. Control the FCS
   - Create/terminate floor
   - Add participant/resource to floor
   - Remove participant/resource from floor



Protocols

- Establish floor control connections between the different entities
  - SIP/SDP (RFC 4583, RFC 5239)
- Coordinate access to shared resources
  - Binary Floor Control Protocol (BFCP)
  - Talk Burst Control Protocol (TBCP)
- Control the FCS
  - Megaco/H.248
  - SIP Floor Server Control Markup Language (SIP-FSCML)

Establish floor control connections between the different entities

- **Examples** of an offer sent by a conference server to a client

```
m=application 50000 TCP/TLS/BFCP *
a=setup:passive
a=connection:new
a=fingerprint:SHA-1 \
   4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
a=floorctrl:s-only
a=confid:4321
a=userid:1234
a=floorid:1 m-stream:10
a=floorid:2 m-stream:11
m=audio 50002 RTP/AVP 0
a=label:10
m=video 50004 RTP/AVP 31
a=label:11
```

Establish floor control connections between the different entities

- **Examples** of an answer returned by the client

```
m=application 9 TCP/TLS/BFCP *
a=setup:active
a=connection:new
a=fingerprint:SHA-1 \
   3D:B4:7B:E3:CC:FC:0D:1B:5D:31:33:9E:48:9B:67:FE:68:40
   :E8:21
a=floorctrl:c-only
m=audio 55000 RTP/AVP 0
m=video 55002 RTP/AVP 31
```

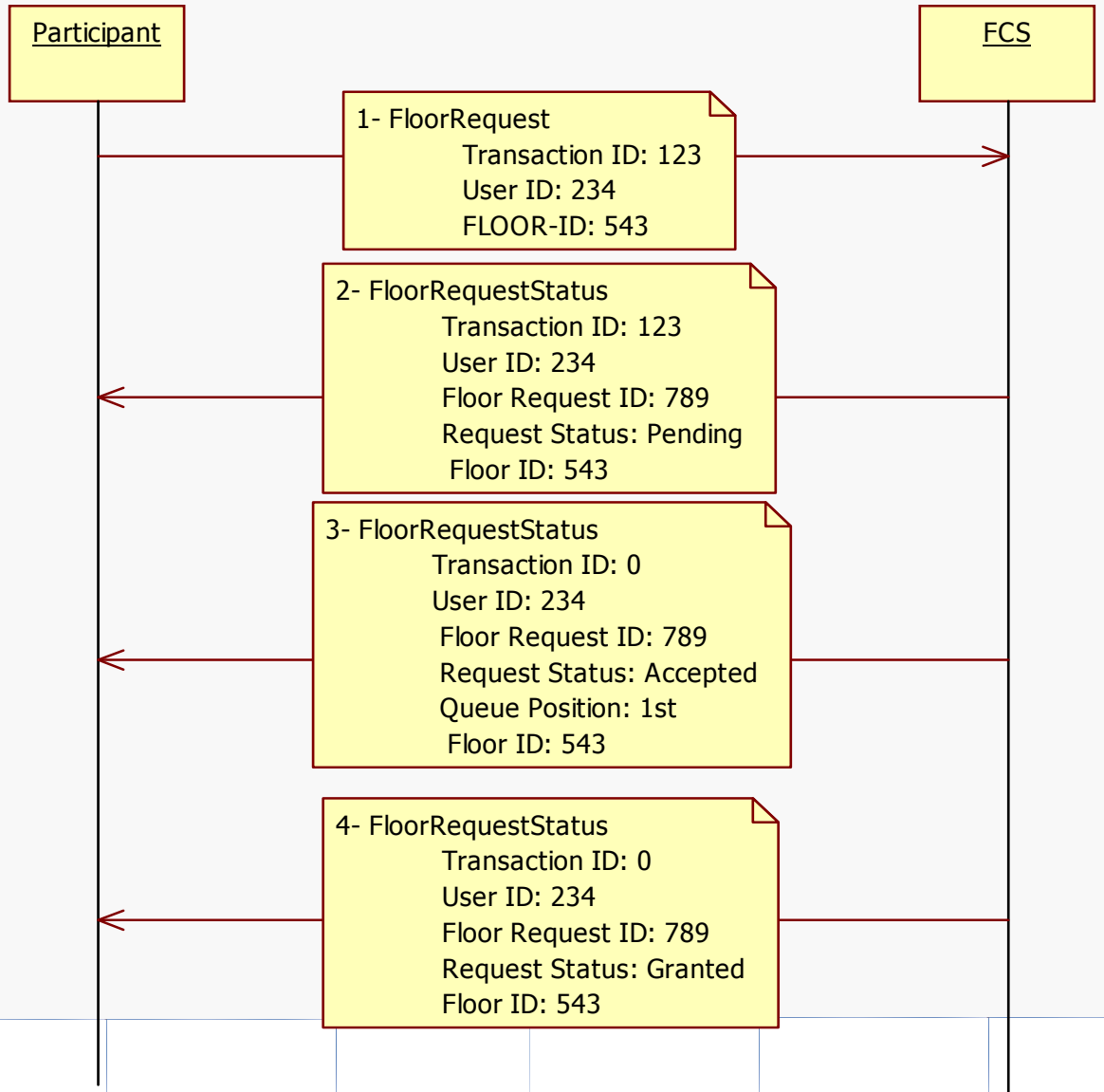
Coordinate access to shared resources

- Binary Floor Control Protocol (BFCP)
  - Standardized in RFC 4582
  - Negotiation of BFCP connections within SIP/SDP
    - Standardized in RFC 4583
  - Advantages
    - Fast (binary encoded)
    - Secure
    - Reliable (over TCP)
    - Provides all the floor control functionalities

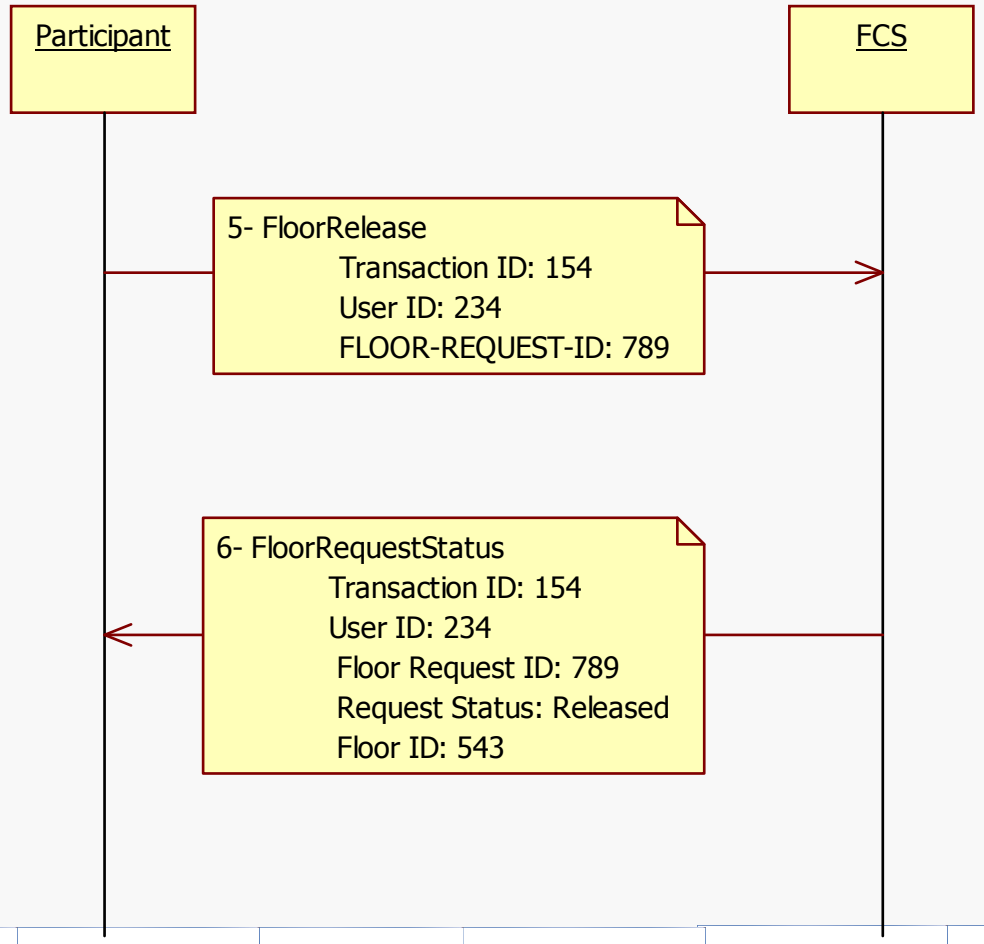
Binary Floor Control Protocol

- Protocol operations and messages/primitives
  - Participant operations
    - Request a floor (FloorRequest)
    - Cancel a floor request (FloorRelease)
    - Release a Floor (FloorRelease)
  - Chair operations
    - Grant a floor (ChairAction)
    - Deny a floor (ChairAction)
    - Revoke a floor (ChairAction)
  - Participant/Chair
    - Requesting Information about Floors (FloorQuery)
    - Requesting Information about Floor Requests (FloorRequestQuery)
    - Requesting Information about a User (UserQuery)
    - Obtaining the Capabilities of a Floor Control Server (Hello)
  - FCS operations
    - Handles the participant and chair requests (FloorRequestStatus, FloorStatus, UserStatus, ChairActionAck, HelloAck, Error)

Binary Floor Control Protocol



Binary Floor Control Protocol

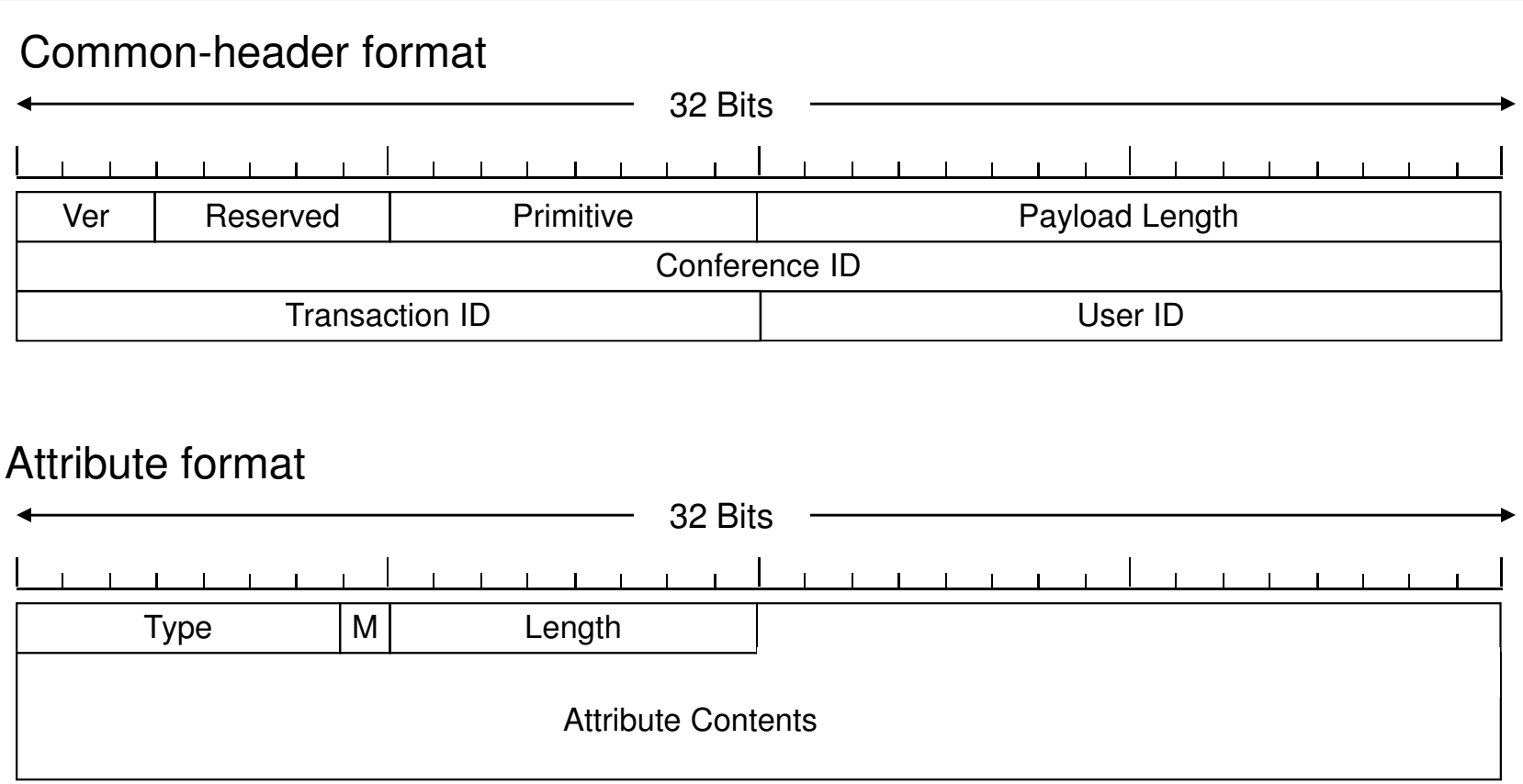


Binary Floor Control Protocol

- Packet Format
  - BFCP messages
    - Consist of a common header followed by a set of attributes.
    - Use a TLV (Type-Length-Value) binary encoding
    - Floor participants, media participants, and floor chairs are identified by 16-bit user identifiers.
    - BFCP supports nested attributes (i.e., attributes that contain attributes).
      - Referred to as grouped attributes.

Binary Floor Control Protocol

- Packet Format



Binary Floor Control Protocol

- Primitives

Value	Primitive	Direction
1	FloorRequest	P -> S
2	FloorRelease	P -> S
3	FloorRequestQuery	P -> S ; Ch -> S
4	FloorRequestStatus	P <- S ; Ch <- S
5	UserQuery	P -> S ; Ch -> S
6	UserStatus	P <- S ; Ch <- S
7	FloorQuery	P -> S ; Ch -> S
8	FloorStatus	P <- S ; Ch <- S
9	ChairAction	Ch -> S
10	ChairActionAck	Ch <- S
11	Hello	P -> S ; Ch -> S
12	HelloAck	P <- S ; Ch <- S
13	Error	P <- S ; Ch <- S

S: Floor Control Server P: Floor Participant Ch: Floor Chair

Talk Burst Control Protocol

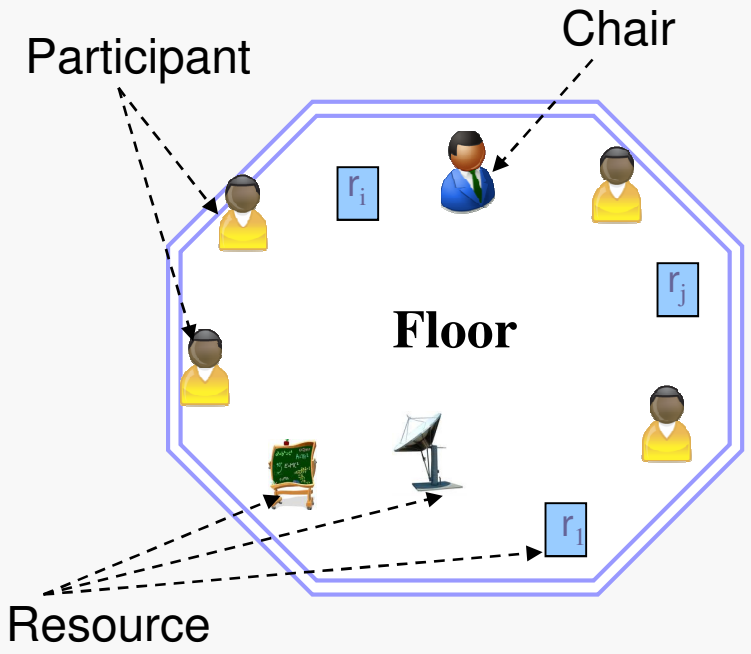
- TBCP
  - Defined by the OMA (Open Mobile Alliance)
  - Uses the application extension features of RTCP (RTP Control Protocol) in order to invoke floor control within the POC (Push to talk Over Cellular) environment.
- Typical TBCP messages include:
  - Talk Burst Granted
  - Talk Burst Request Message
  - Talk Burst Deny Message
  - Talk Burst Release Message
  - Talk Burst Taken
  - Talk Burst Idle
  - Talk Burst Revoke
- Advantages
  - Fast
  - Secure
- Disadvantages
  - Only provides basic floor control functionalities (e.g. no chair supported).

Control the FCS

- SIP-FSCML is a non-standard alternative to H.248
  - Less complex
  - Easy to understand and use by SIP application developers.
- It follows SIP and XML paradigms.
- It enables a peer-to-peer communication model between the AS and the FCS.
  - This allows the FCS to be simultaneously used by multiple ASs.

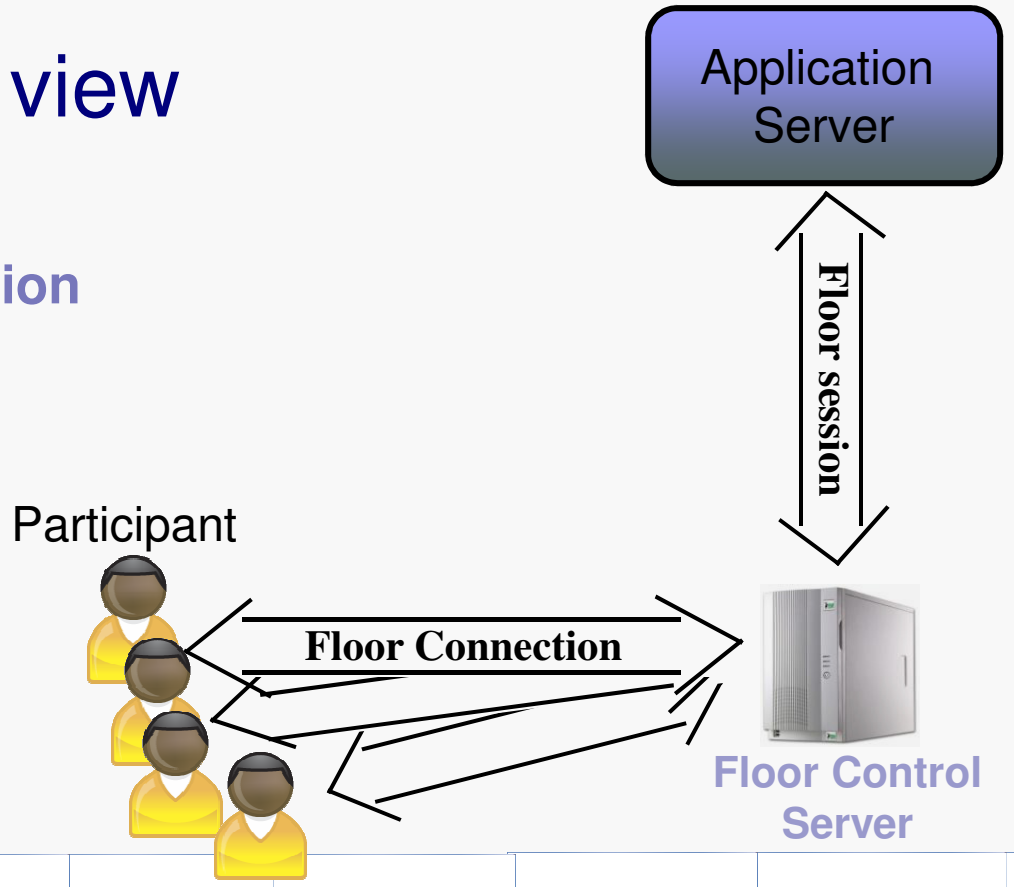
SIP Floor Server Control Markup Language

- Conceptual view
  - Floor
  - Floor Connection
  - Floor Session



SIP Floor Server Control Markup Language

- Conceptual view
  - Floor
  - Floor Connection
  - Floor Session



SIP Floor Server Control Markup Language

- The control session between the application and the FCS is opened through a SIP INVITE message.
- FSCML requests to the FCS are carried in SIP INFO messages
  - Each INFO message includes a single FSCML body
  - An FSCML body can carry any number of FSCML requests
- SIP-FSCML responses are transported in a separate INFO message
- SIP-FSCML Is a request-response protocol; with only final responses
- SIP-FSCML relies on SIP subscribe/notify mechanism, to allow applications subscribe to floor control related events

SIP Floor Server Control Markup Language

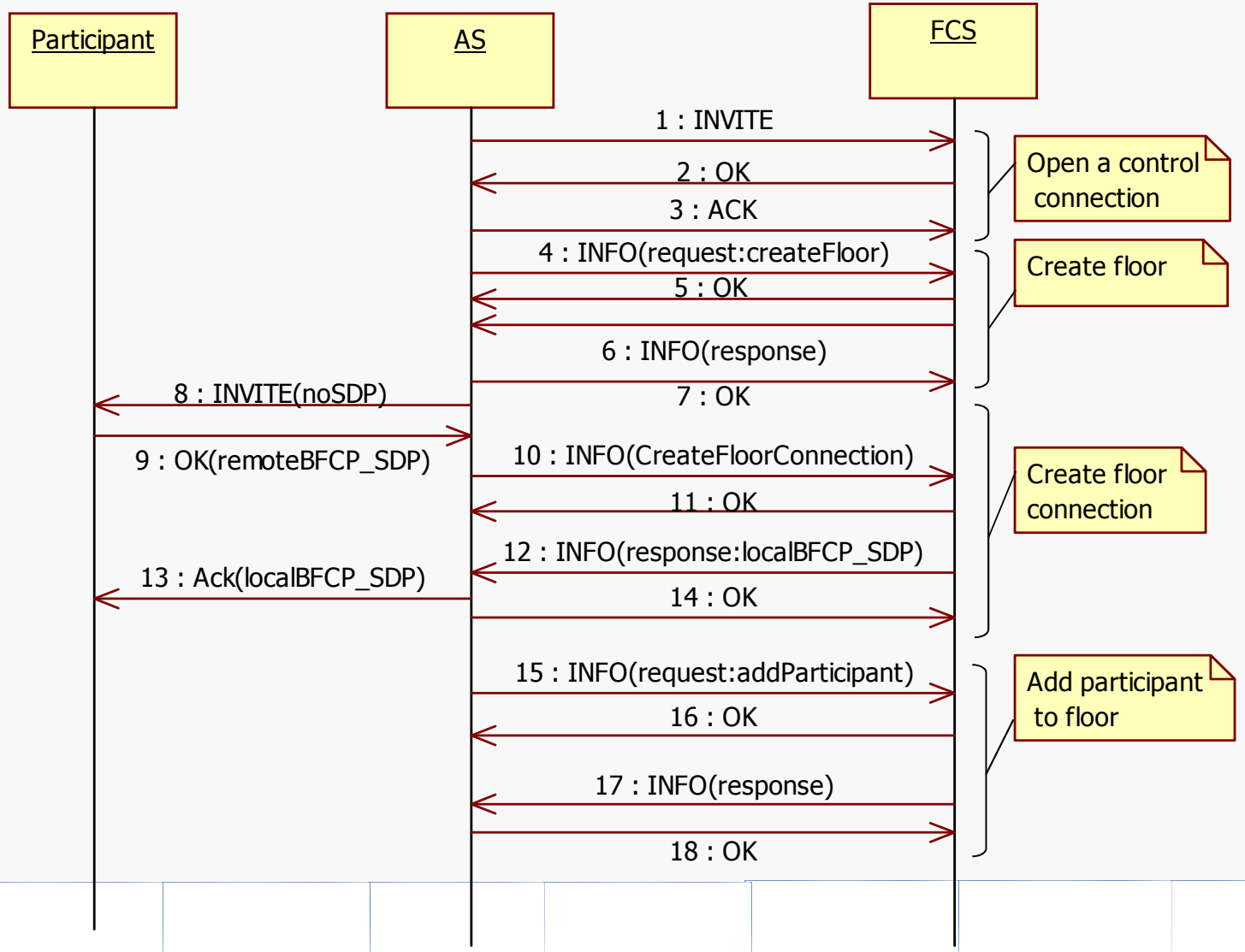
- SIP-FSCML operations
  - Open/close control connection
  - Create floor
  - Create floor Connection
  - Add/remove floor to/from a conference
  - Set/update Chair for a floor
  - Add/remove floor participant(s)
  - Set floor algorithm
  - add/remove media to/from a floor
  - Set maximum floor holders
  - Set maximum floor holding time

SIP Floor Server Control Markup Language

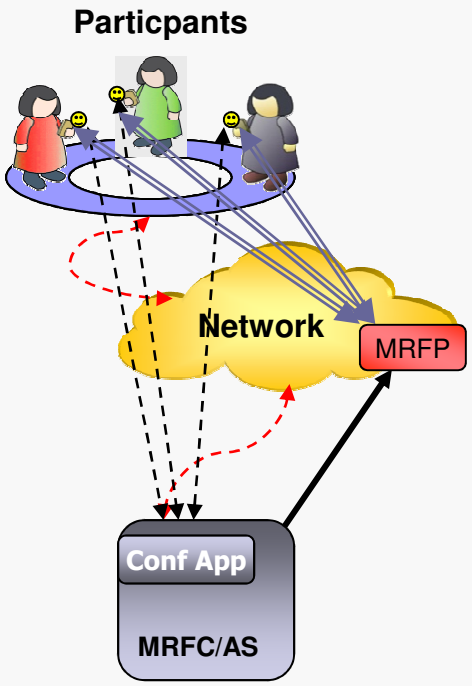
- Example of FSCML body

```
<FloorServerControl>
 <conferenceid>the conference ID</conferenceid>
 <request type=" CreateFloor">
   <floorid>the floor ID</floorid> (mandatory)
   <algorithm>the floor control algorithm</algorithm> (mandatory)
   <maxholders>max number of floor holders </maxholders>(optional default=1)
   <maxholdingtime>max time (in seconds) a participant can hold a floor, in case someone else
                   asked for it          </maxholdingtime>(optional, default 0=unlimited)
 </request>
 <request type=" SetChair">
   <floorid>floor whose chair should be set</floorid> (mandatory)
    <chairid>the chair ID</chairid>
 </request>
 <request type=" AddParticipant">
   <floorid>id of the floor to which to add</floorid> (mandatory)
   <participantid>the participant ID</ participantid > (mandatory)
 </request>
</ FloorServerControl>
```

SIP FSCML- A Scenario



Putting it together



- Architecture
- Server side scenario
- Client side scenario

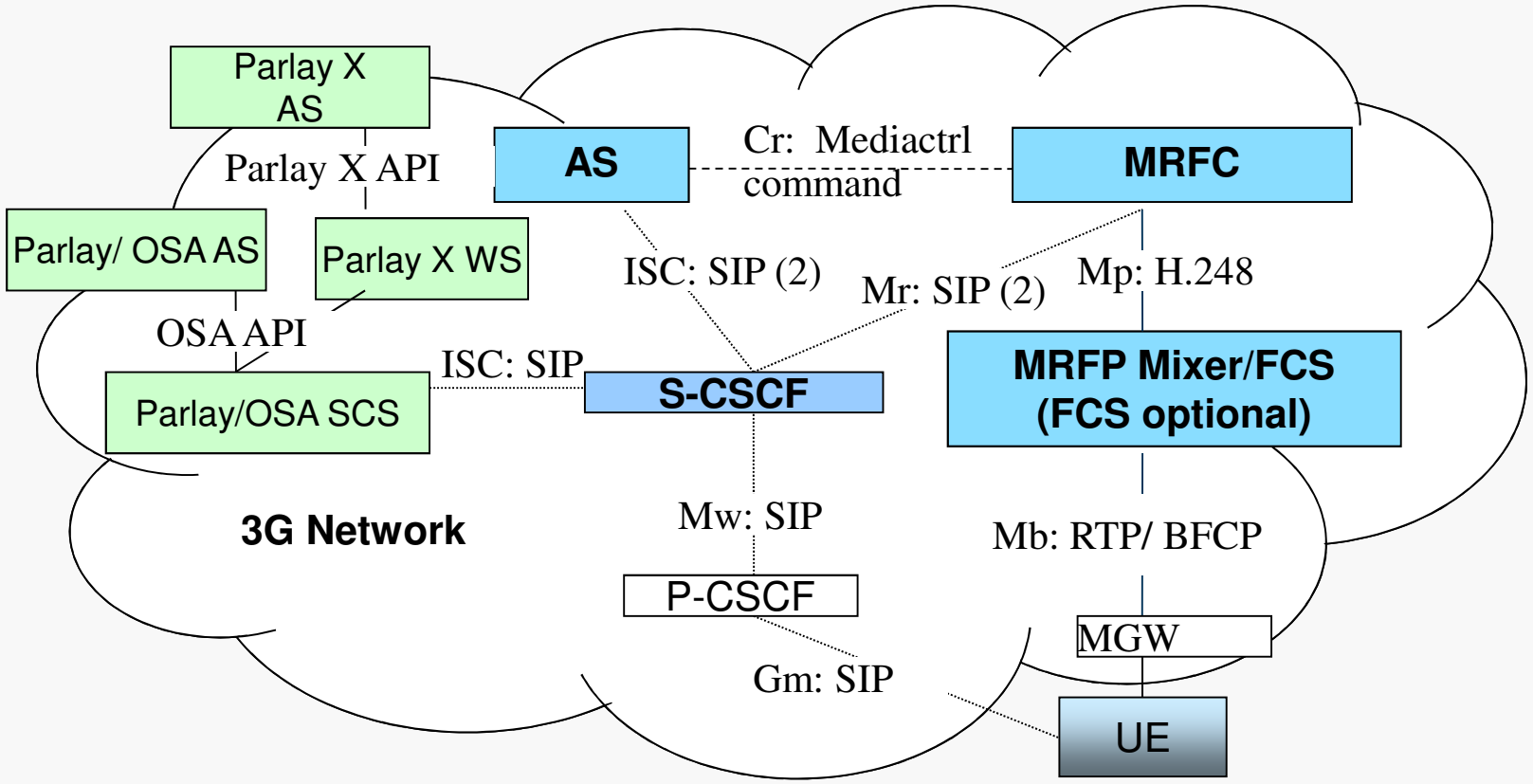
Architecture

- 3GPP conferencing architecture
- Extension to 3GPP conferencing architecture

3GPP conferencing architecture

- The 3GPP IMS conferencing architecture uses the tightly-coupled (centralized) conference model
- Conferencing technical components
  - Signaling
    - SIP
  - Media control
    - H.248
  - Floor control (optional)
    - Floor control connections' establishment
      - SIP/SDP
    - Coordination of access to shared resources
      - Binary Floor Control Protocol (BFCP)
    - Control of the FCS
      - H.248

3GPP conferencing architecture

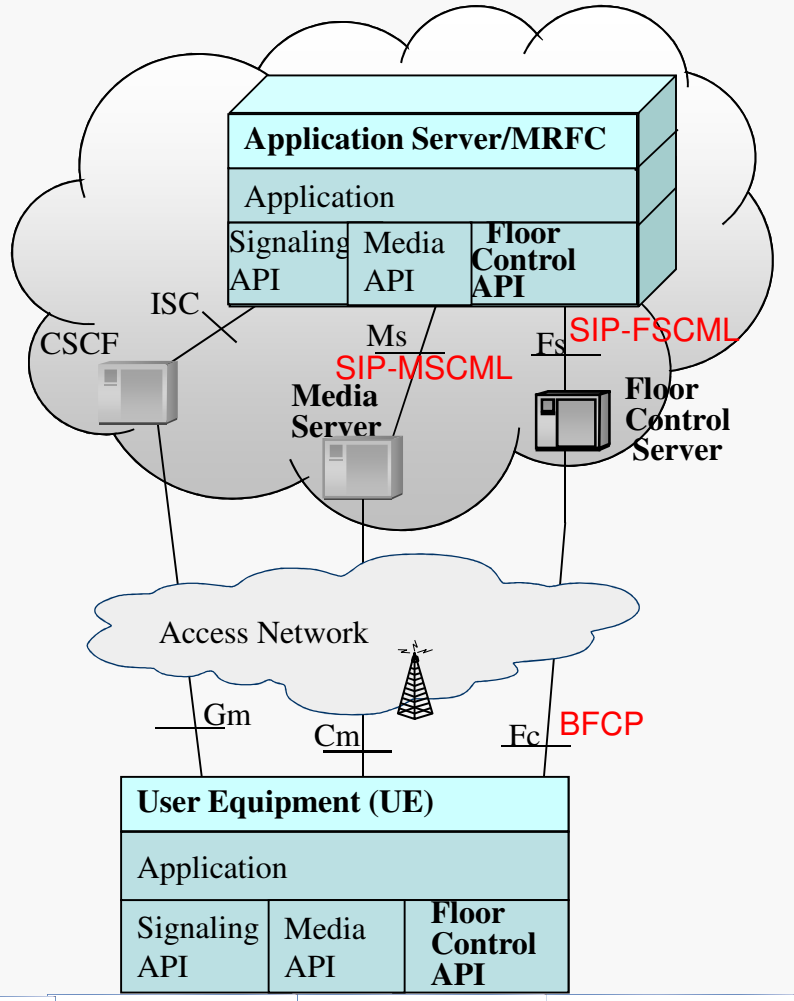


3GPP conferencing architecture

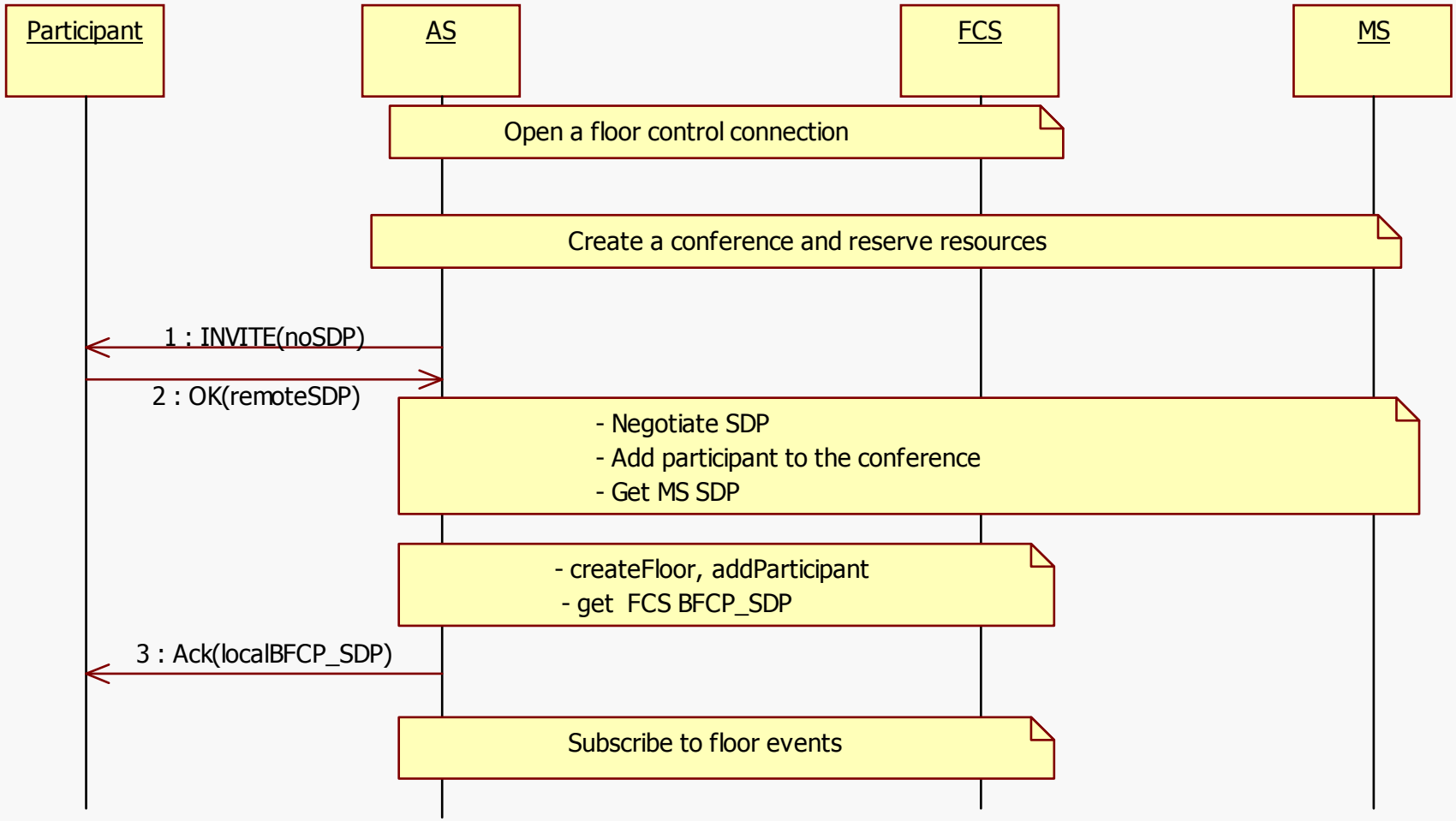
- Limitations
  - The FCS is located inside the MRFP
    - The MRFP has to host a brand new functional entity
  - There is no interface between the two MRFP and the FCS
    - MRFP and floor control node have to be bought from the same supplier
  - No API is provided for application development

Extension to 3GPP conferencing architecture

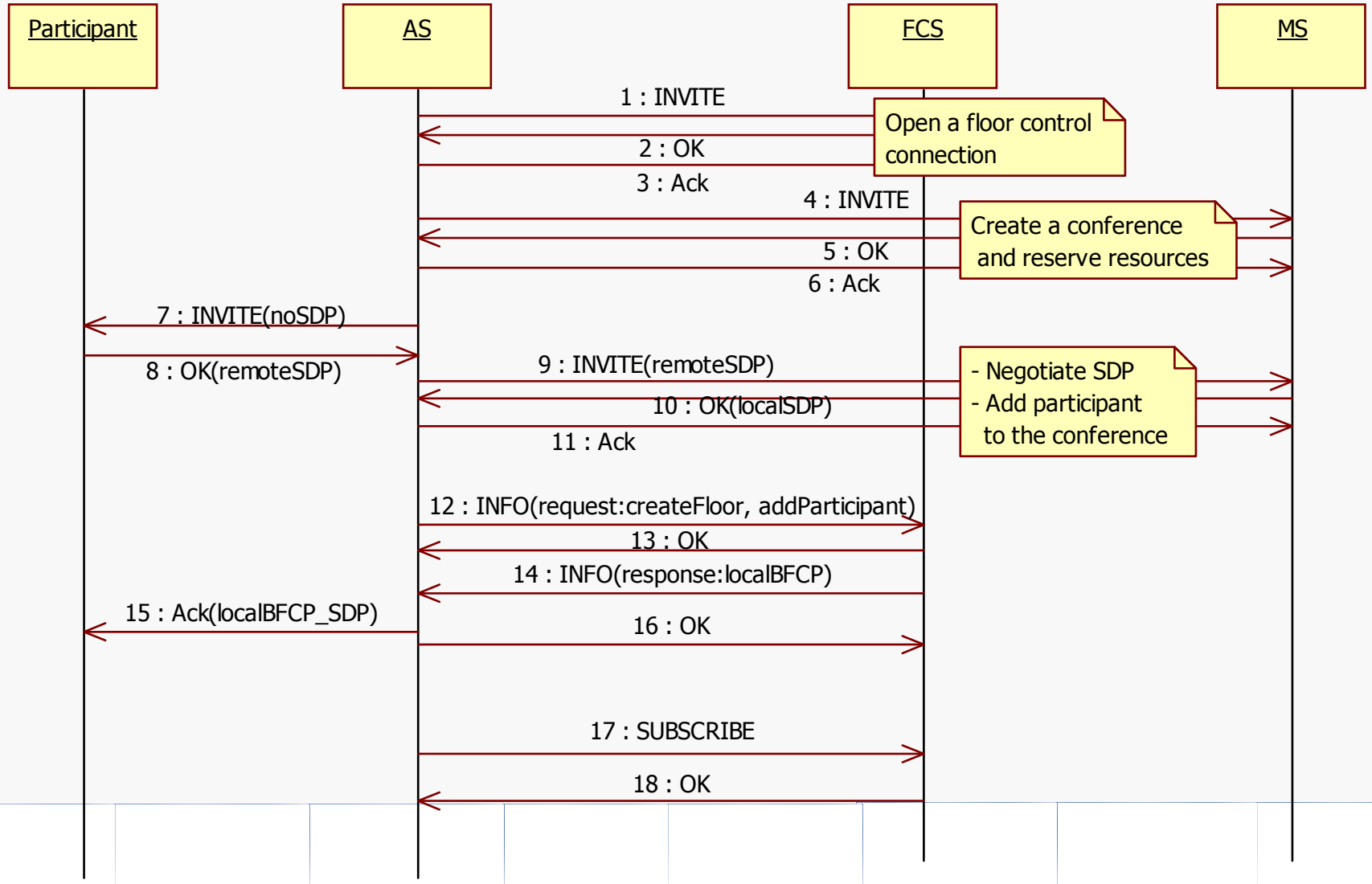
- Separate FCS from media server
- Enables the use of emerging industry standards such as SIP-MSCML for Media server control.
- Uses SIP–FSCML as an alternative to Megaco/H.248 for FCS control
- Includes a comprehensive set of server side and client side API that exposes the floor control capabilities



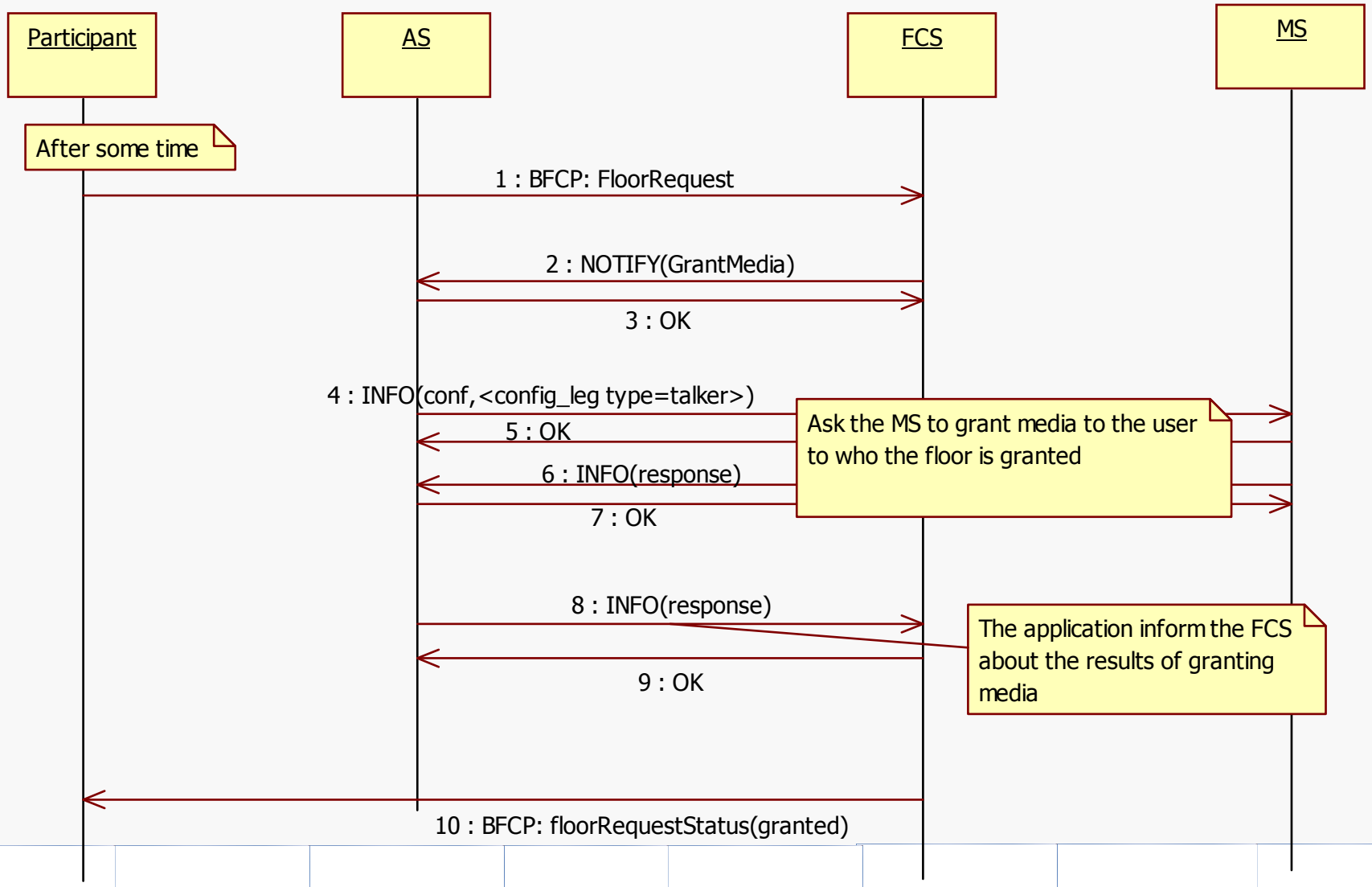
Server side scenario



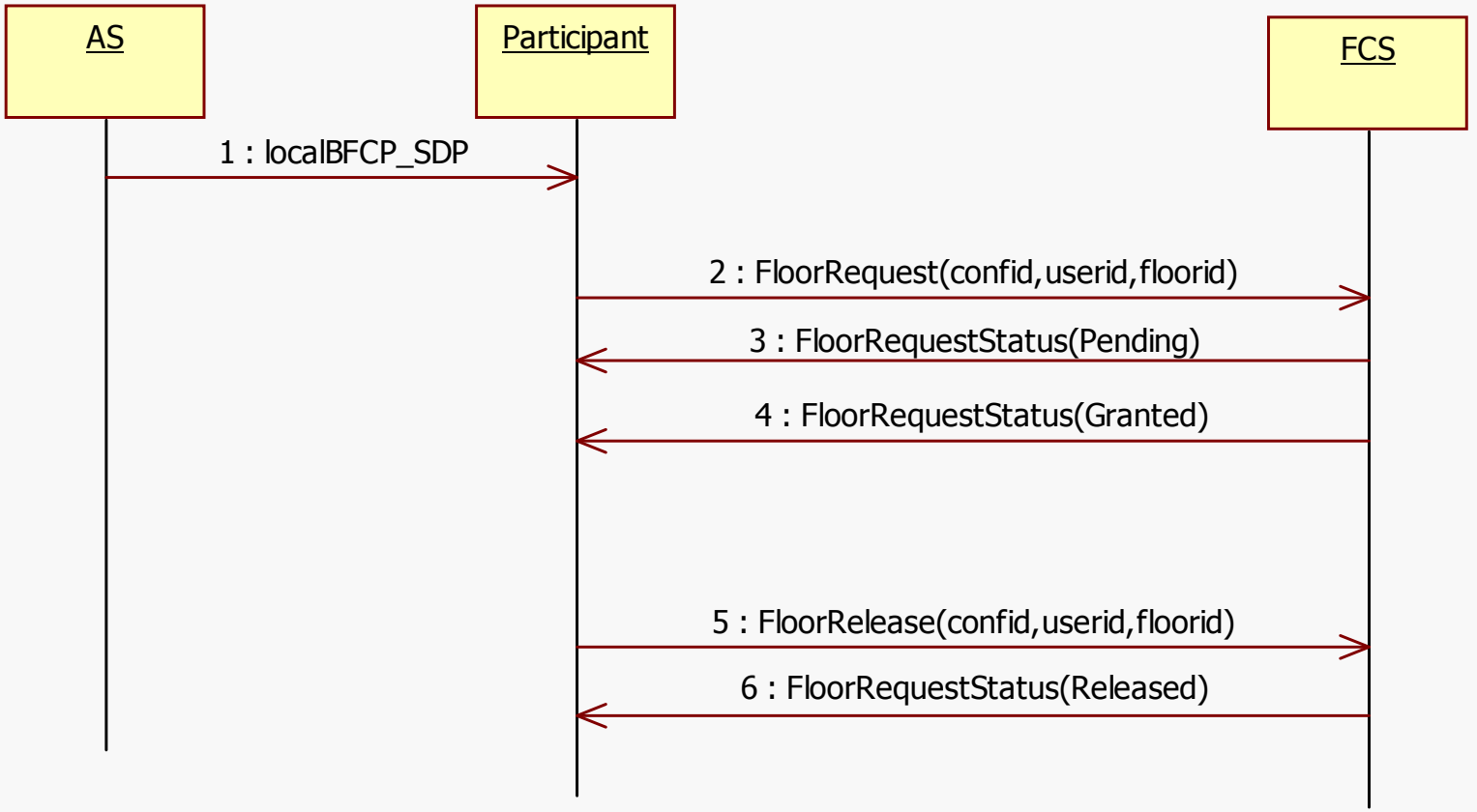
Server side scenario



Server side scenario



Client side scenario

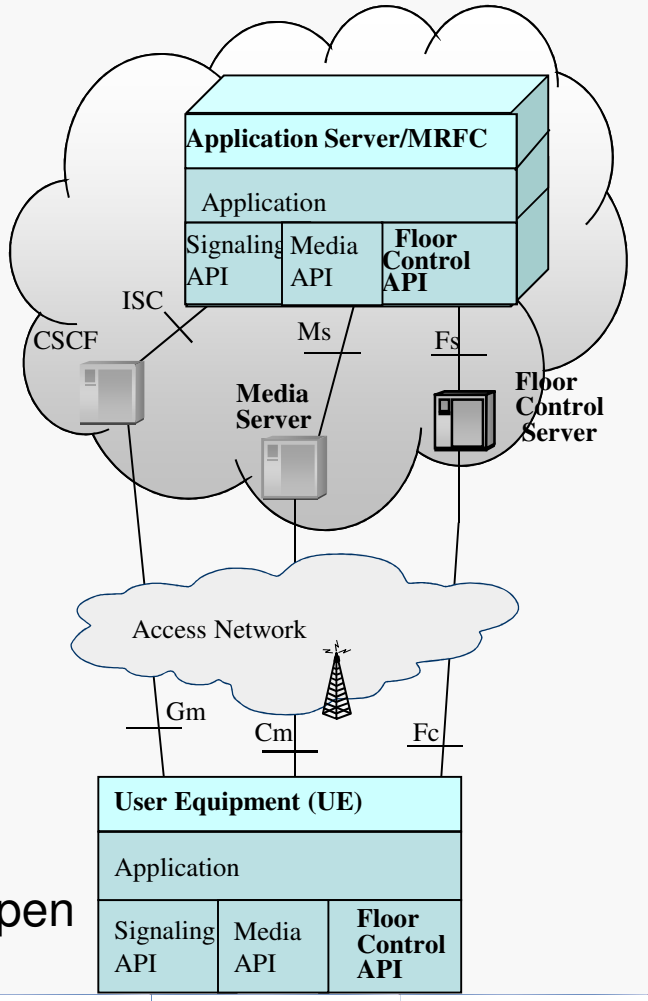


Case study

- Multiparty Multimedia applications' development
- Game semantics
- Implementation architecture
- Prototype
- Performance evaluation

Conferencing applications' development

- APIs
  - Standard APIs
    - 3GPP & Parlay forum
      - Parlay/OSA
      - Parlay X web services
    - Java Community Process (JCP)
      - A set of JSRs
        - E.g. JSR 289 (SIP Servlet), JSR 309
  - Non standard APIs
    - E.g. A comprehensive set of server side and client side floor control APIs



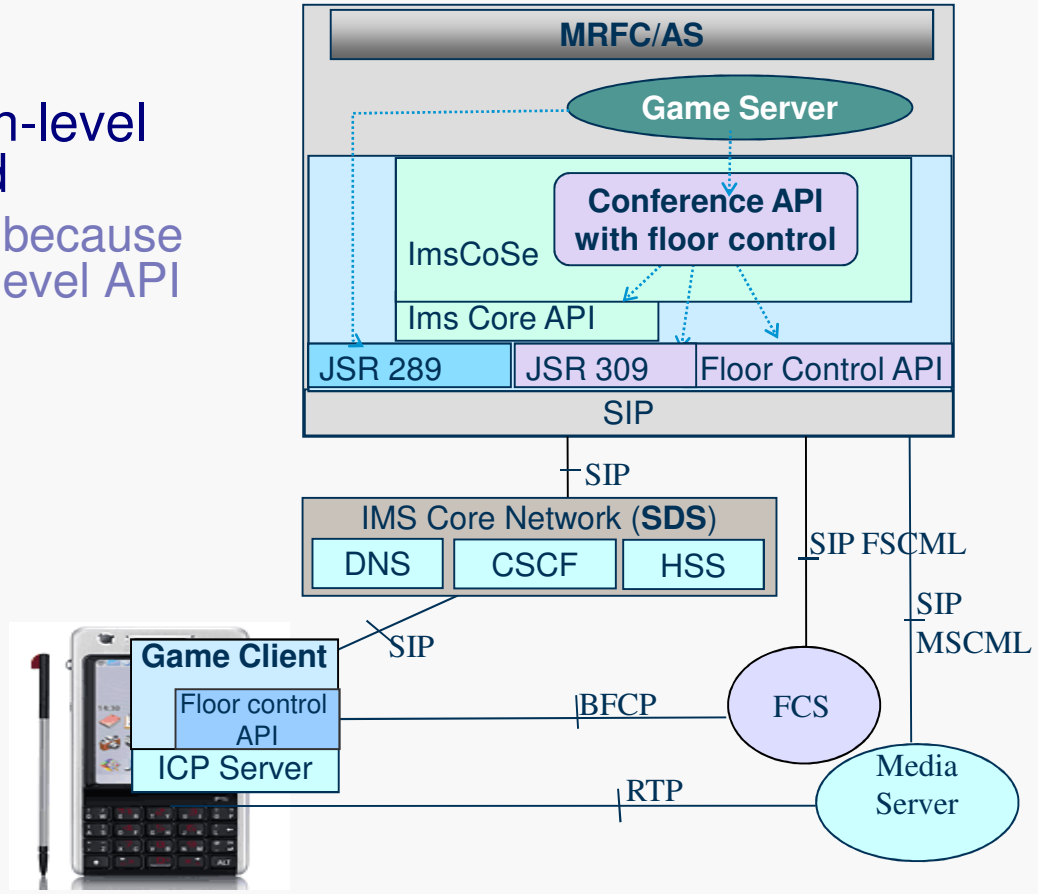
- Tool kits
  - Ericsson Service Development Studio (SDS)
  - Open IMS Core from Fraunhofer Institute for Open Communication Systems (FOKUS)

Game semantics

- "Capture the Flag" is
  - A multi-party game
  - Played by 2 teams
  - Each team has a base and a flag inside the base
  - Teams fight to kill the opponents and capture their flag
  - The first team that captures the flag wins the game
- Revised "Capture the Flag"
  - Players can communicate
  - Each team has one chance to bomb a zone, i.e. kill everybody within the range except the one that bombs
  - Each team has a captain (which can manage talking and bomb floors)
  - The team should at least kill one opponent before capturing the flag
  - Some announcements are played during the game
    - E.g. start game, enemy killed, game over

Implementation architecture

- The game server uses a high-level conference API we designed
  - We have designed this API because of the unavailability of high level API for conferencing in IMS.
  - The API is built on:
    - JSR 289
    - JSR 309
    - floor control API
- The game client uses ICP API and the client side floor control API.



Implementation architecture

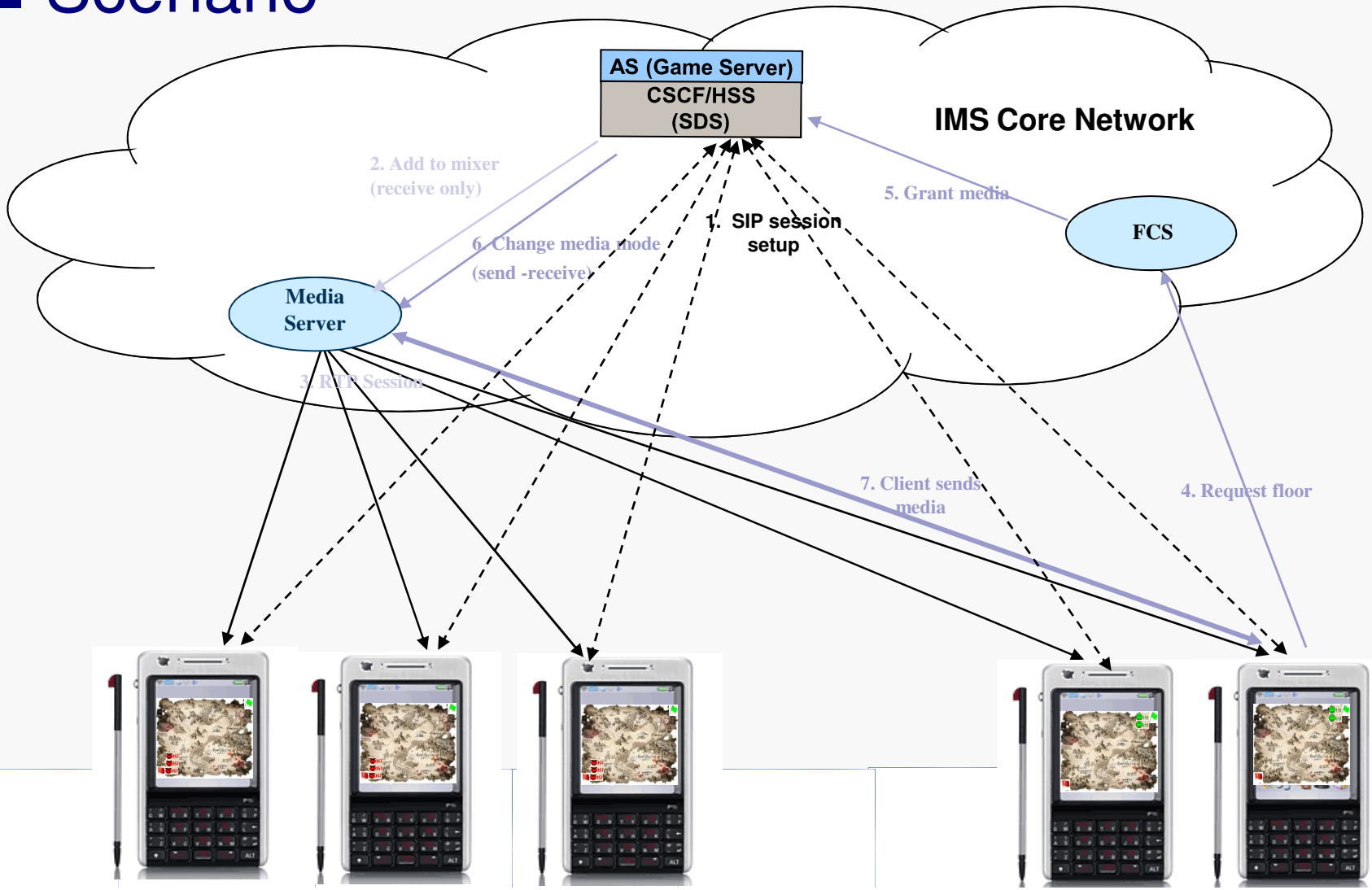
- There are two types of interfaces between the client and the server:
  - Game control interface
    - The game control messages are carried in the body of SIP Message in an XML format.
    - The game control messages:
      - startGame,
      - playerMove
      - zoneUpdate
      - shoot
      - Bomb
      - shoot and bomb responses
      - enemyDead
      - capture
      - gameOver
  - Conferencing interface
    - INVITE, BYE, …

startGame Message

```
<startGame>
        <maxBullets> 6 </maxBullets>
        <localPlayerId>123</
        localPlayerId>
        <teamPlayers>
                <playerId>456</ playerId>
                <playerId>356</ playerId>
        </teamPlayers>
        <teamId>Angels</teamId>
        <zoneId x=0, y=0 />
</startGame>
```

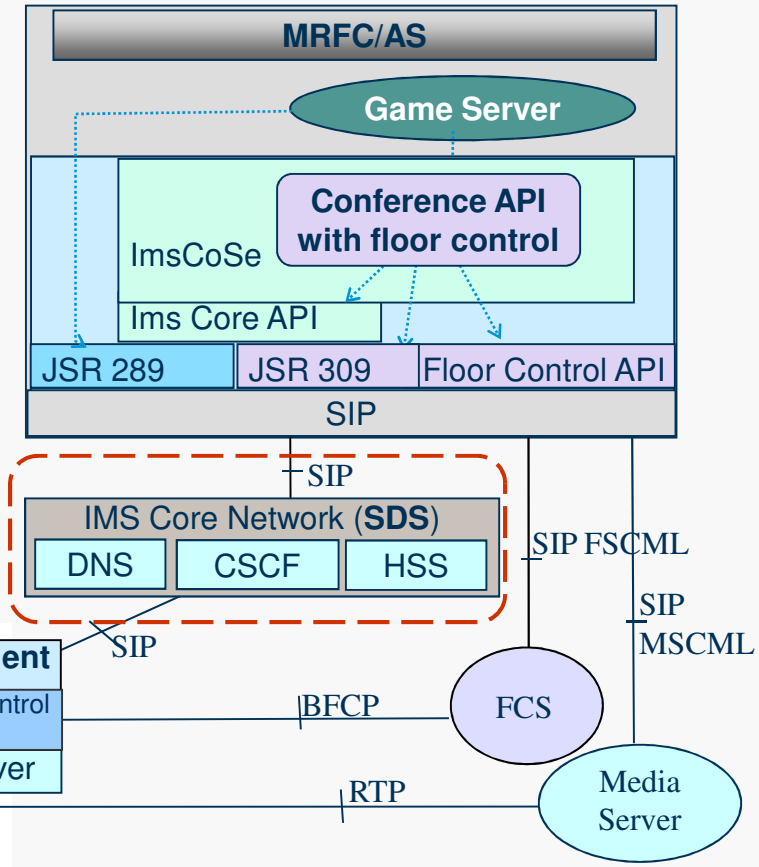
Implementation architecture

- Scenario



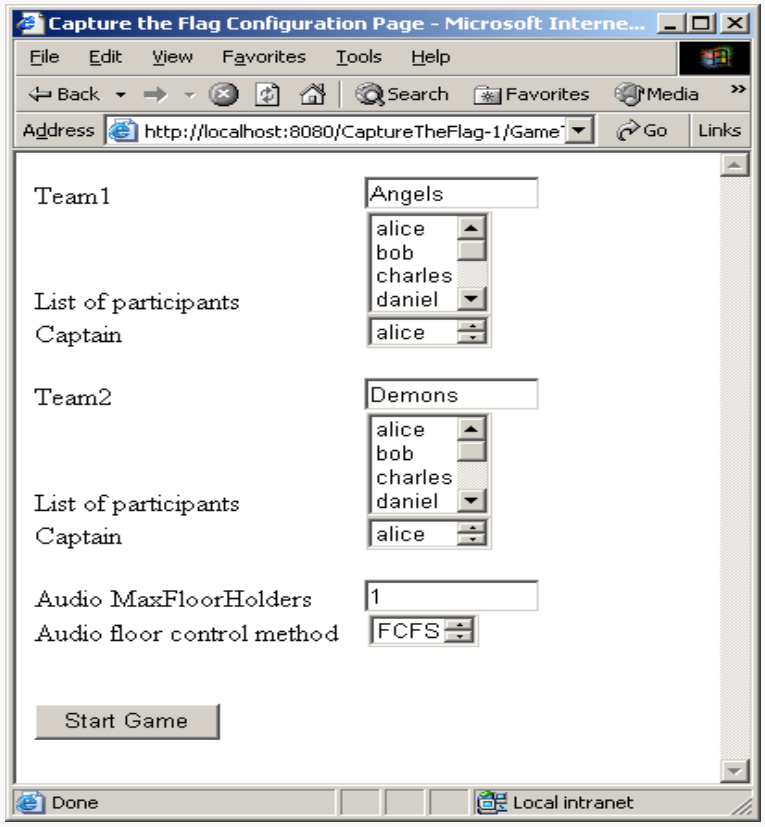
Prototype

- Network nodes
  - IMS network: simulated using SDS
  - SIP AS: GlassFish/Sailfin
  - FCS implement in a previous work
  - Media server: MRFP prototype from Ericsson
- APIs
  - We implemented the high-level conference API using a subset of JSR 309 implementation
  - Reused the server-side floor control API implemented before
  - We used the SDS implementation of ICP API
- The game server is deployed as a SIP Servlet in the AS

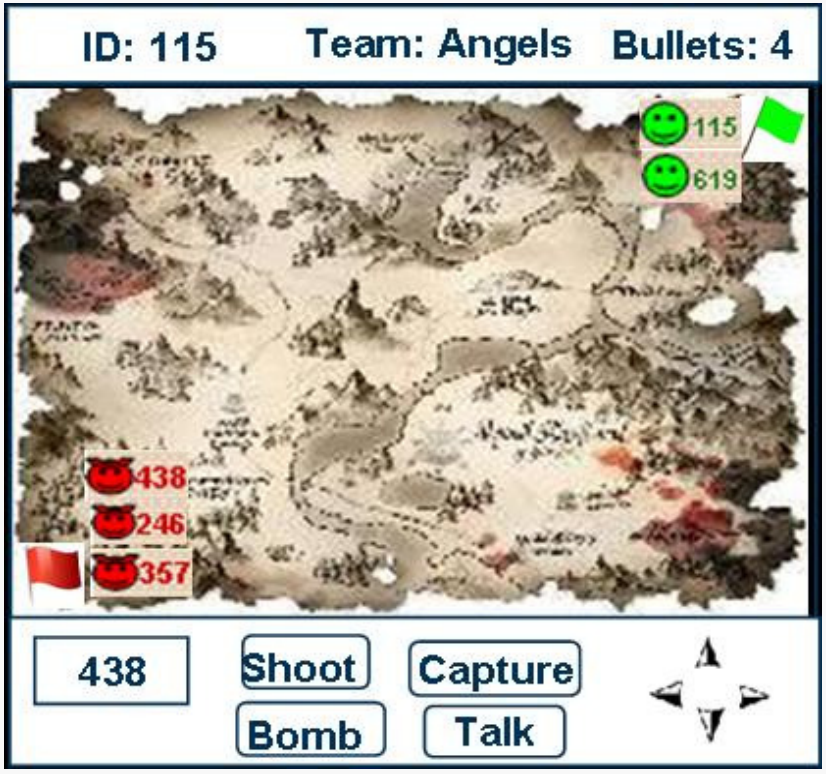


Prototype

- Server GUI

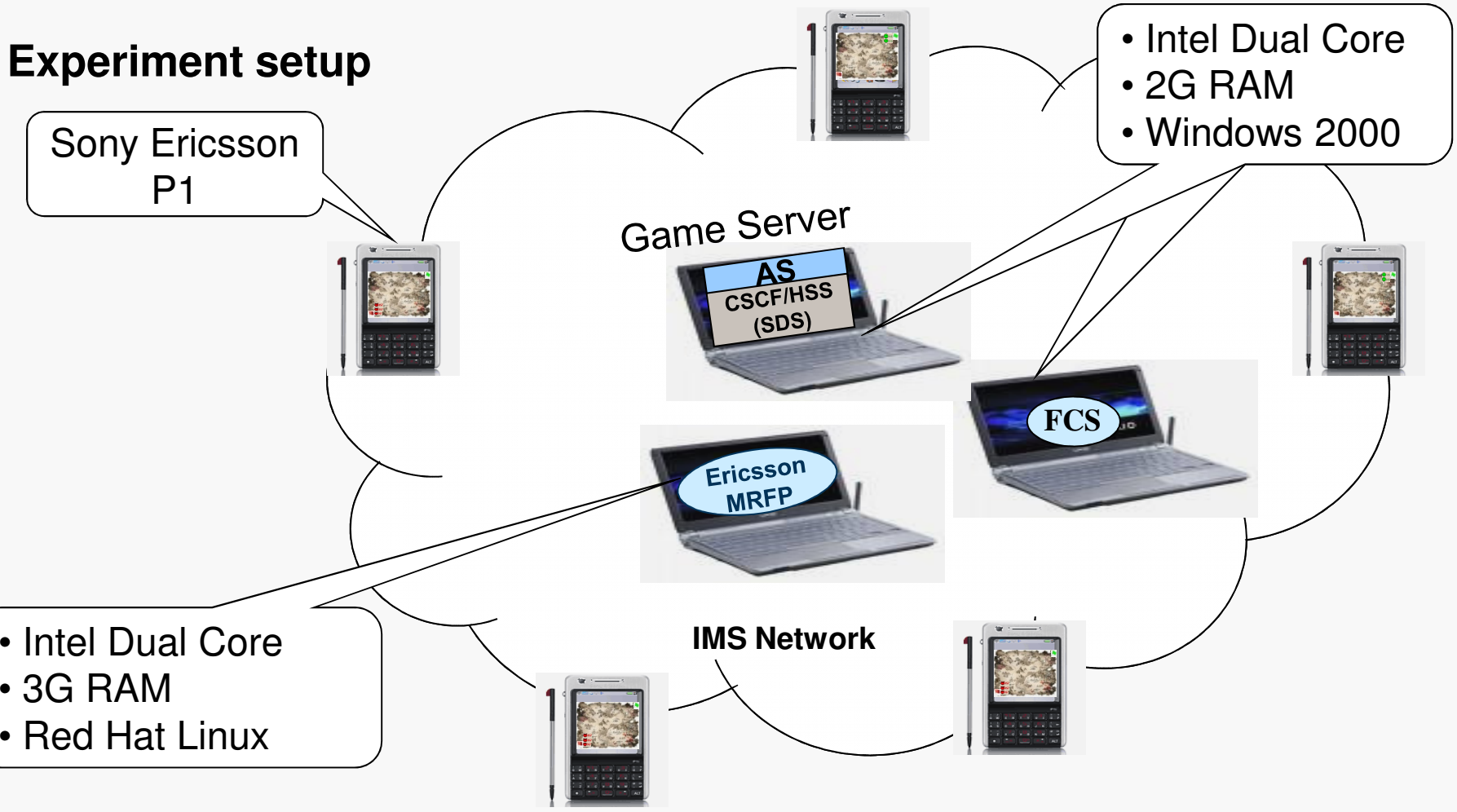


- Client GUI



Performance evaluation

Experiment setup



Performance evaluation

Conference establishment code footprint: with and without using IMS high-level API

JSR 289+JSR 309	high-level IMS API
> 20 lines	~ 3 lines

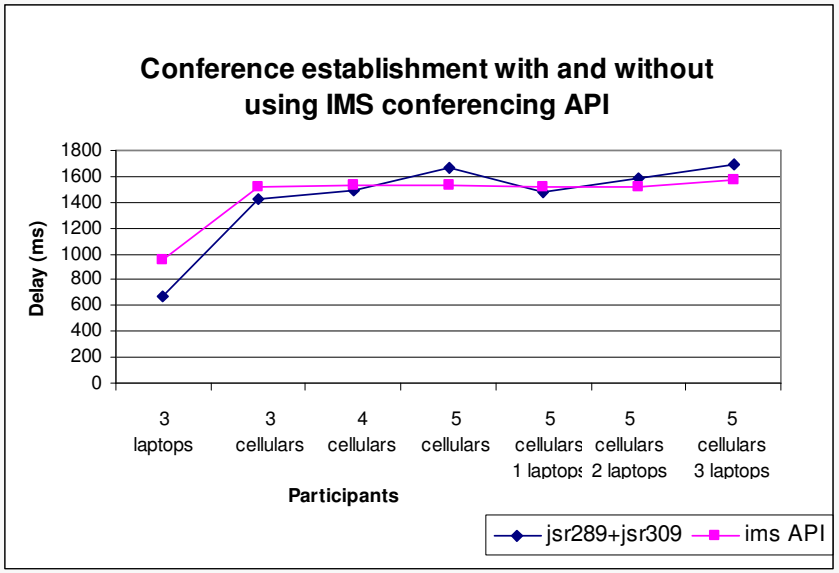
```
// create conference service
 ImsConferenceService confService = ImsServiceManager.getInstance().createService(id, "Conferencing");

// create conference policy
 ImsConferencePolicy confPolicy = service.createConferencePolicy(startTime,duration, maxParties,..);

// create a conference with participants
ImsConference conference = confService.createConference(confPolicy, partyList);
```

Performance evaluation

Conference establishment delays: with and without using IMS high-level API



Average Game start and stop delay in server side

Delay (ms)	3 laptops	3 cell phones	4 cell phones	5 cell Phones	5 cells +1 laptop
Game Start delay	1439	2902	3311	3786	3057
Game Stop Delay	151	900	757	868	963

Average game response time in client side

Response Time (ms)	Talk	Req Bomb	Shoot	Capture	Bomb	Move
Laptop	1746	1810	133	172	178	31
Cellular	2368	3001	1008	2453	1661	902

References

- G. Camarillo et al., “The Binary Floor Control Protocol (BFCP)”, RFC 4582, November 2006.
- Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", RFC 4583, November 2006.
- M. Barnes, C. Boulton and A. Levin, "A Framework and Data Model for Centralized Conferencing", RFC 5239, June 2008
- F. Belqasmi, C. Fu, M. Alrubaye, R. Glitho, “Design and Implementation of Advanced Multimedia Conferencing Applications in the 3GPP IP Multimedia Subsystem”, IEEE Communications Magazine, forthcoming, Octobre 2009
- C. Fu, F. Belqasmi, M. Alrubaye, R. Karunamurthy, R. Glitho, "A Case Study on Multiparty Multimedia Game Development in the IP Multimedia Subsystem”, International Conference on Intelligence in Networks (ICIN 2009), Forthcoming, October 26 – 29, 2009
- F. Belqasmi, M. Al Rubaye, C. Fu, R. Glitho, “A Novel Architecture for Floor Control in the IP Multimedia Subsystem of 3G Networks”, IEEE Vehicular Technology Conference (IEEE VTC2009-Spring), April 26-29, Barcelona, Spain