Advanced Multimedia System (AMS)

Overview and Update

Paul E. Jones

Rapporteur, ITU-T Q2/16

7 January 2014

Outline

- Introduction
- User Scenarios
- Architecture
- Protocol Overview

Introduction

AMS: The Third-Generation System

- The Advanced Multimedia System (AMS) is a new multimedia communication system under study by ITU-T SG16
- It is the third such multimedia system developed by the ITU-T, with H.320 and H.323 systems representing the first two generations
- When complete, the base specification will be published as H.325

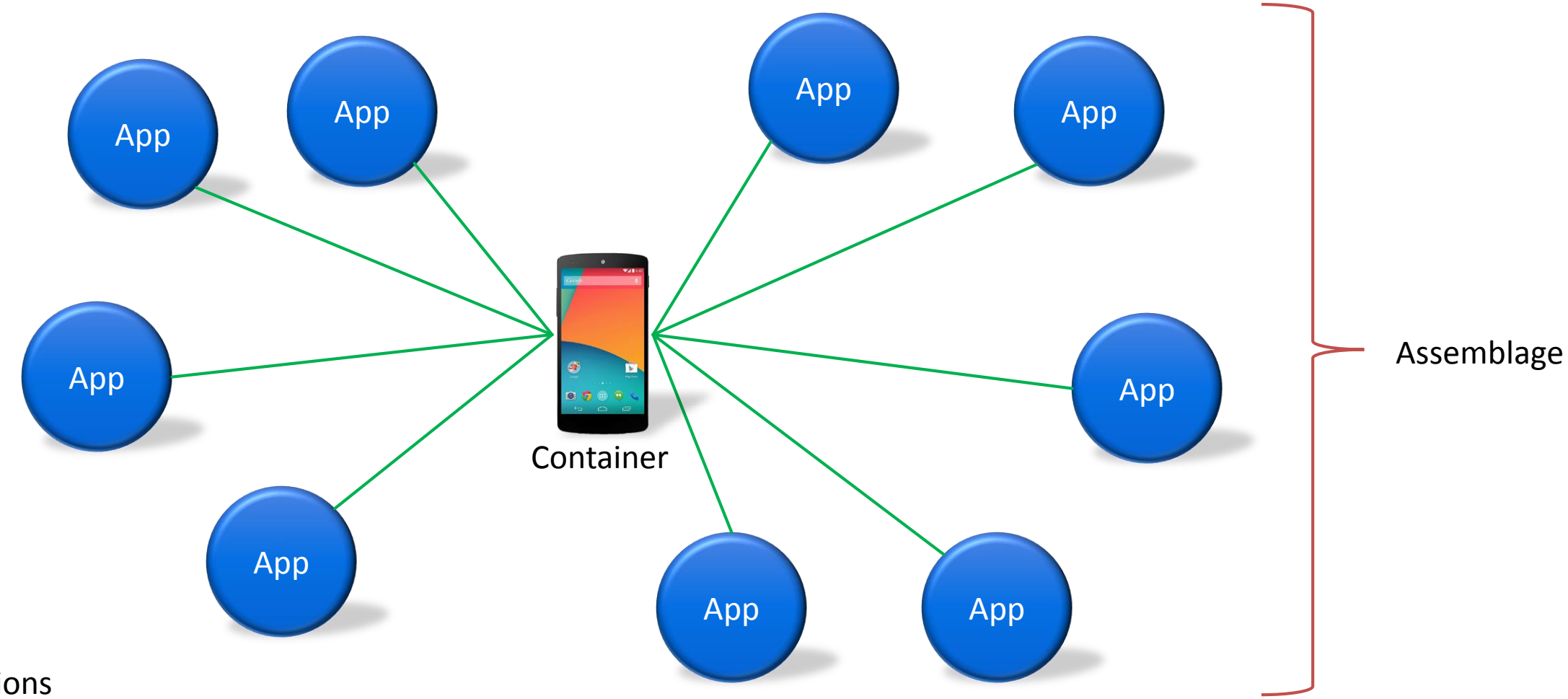
Why AMS is Important

- There is a desire to enable end users with the ability to utilize any number of different applications simultaneously
  - Applications might run on the same or different devices
  - Applications might be implemented by different vendors
- Accessibility considerations are important (e.g., flashing lamp)
- Legacy systems have been narrowly focused on voice and video, with data applications considered secondary
  - AMS will put all applications on equal footing
  - Application integration is an integral part of the architecture
- Realization of the “any device, anytime, anywhere” concept

Basic Terminology

- Container
  The entity, such as a software application or application device, with which the user interacts to control sessions, associate applications, etc.
- Application
  Entities that implement real functionality for the user, such as a voice/video app, file transfer, or electronic whiteboarding
- Service Node
  An element in the network with which Containers and applications register, facilitating exchange of messages between Containers and applications, and providing assistance with NAT/FW traversal (via integrated or external traversal servers)
- Association
  A relationship established between a Container and an application
- Assemblage
  The Container and all of its associated applications
- Multipoint Control Unit
  A special application, often implemented in a dedicated network element, that provides n-way conferencing functionality for multiple users and for application that are not intrinsically multipoint capable to effectively communicate in a multipoint conference
- Gateway
  An application often implemented in a dedicated network element that enables communication with legacy network protocols, such as PSTN, H.323, H.320, or SIP

The Container and Associated Applications



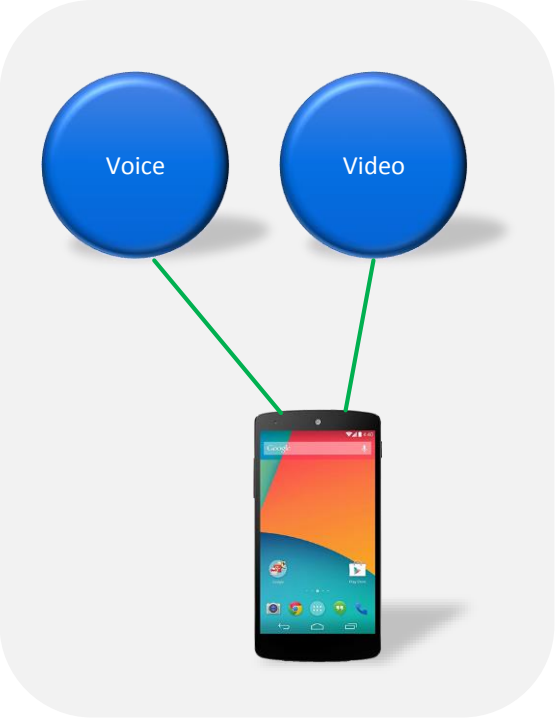
Applications are Everywhere

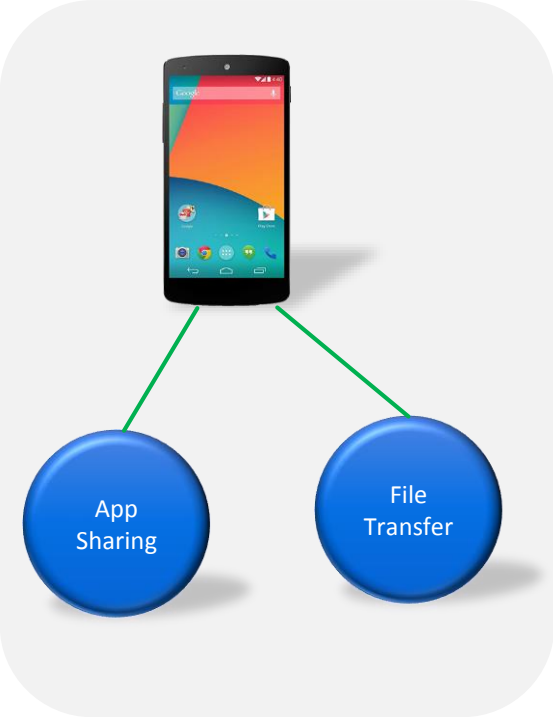
Applications are Everywhere (in words)

- Some applications may be integrated with the Container, such as voice, video, and file transfer
- Many more applications may be used that exist outside the Container, including
  - External monitors for presentation of video
  - Monitors in the back of a taxi
  - Table-top interactive displays
  - Intelligent room-based whiteboards
  - Applications installed on a computer
  - Desktop telephones
  - Video game systems

Application Sets

- Application Sets are a logical grouping of applications that the user would like to use during a communication session
  - Some applications may be configured to invoke upon session establishment
  - Some applications might be the default application for a given context
- There may also be different application sets defined for work or home
- This is strictly a user interface behavior in the Container, not something defined in the protocol





“Moving” Applications (a.k.a., “Application Handover”)

- Multiple instances of the same kind of application may be registered with the Container (e.g., multiple “videoconferencing” applications may be at the user’s disposal)
- A user may “move” the “videoconferencing” application instance from a mobile handset to a PC or telepresence system, for example, without transferring or otherwise disrupting the other communicating applications
- “Move” operations may be performed manually or may be automated, depending on the capabilities of the Container and environmental stimuli

User Scenarios

File Transfer Scenario

Alice is working from home sitting at her computer, equipped with an AMS Container and voice/video on a physical phone, along with a file transfer app and application sharing app installed on the computer

Bob is in his car on the way to the office, with a mobile communicator in his hand equipped with a Container and voice/video app. The Container is also associated with apps running on his office computer.

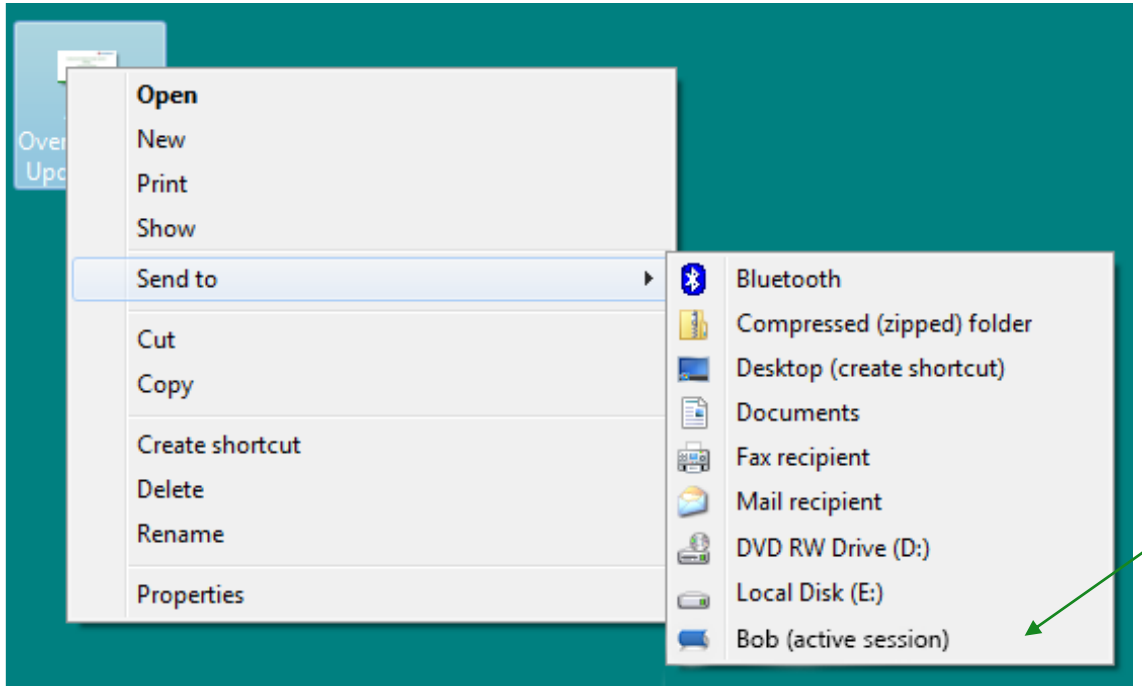
At the office, Bob has a computer and phone equipped similarly to Alice

File Transfer Scenario (cont.)

❶ Alice initiates a voice call to Bob. During the conversation, Alice request that he look over a presentation she prepared.

❷ Bob agrees and asks her to send the file over.

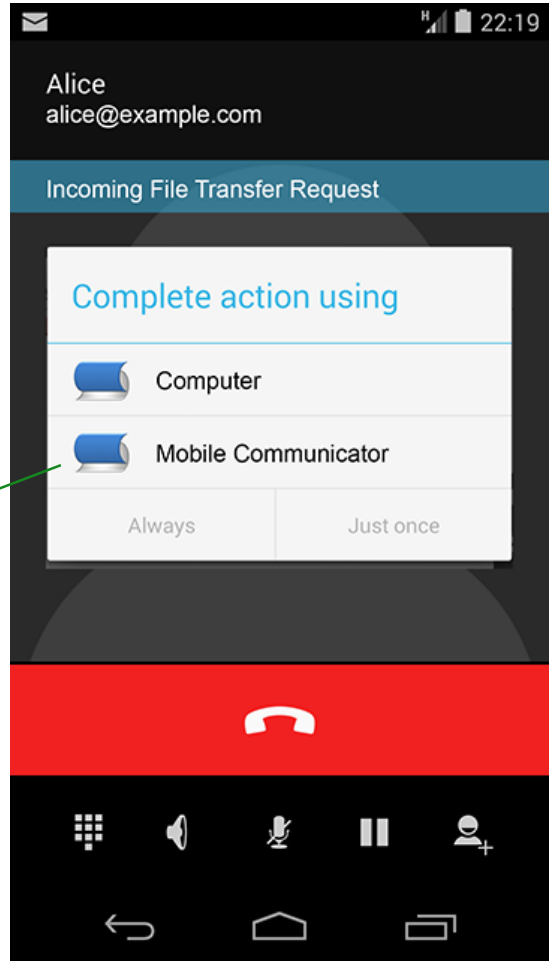
❸ Alice selects the file to send to Bob, as shown in the picture below.



Alice's computer knew about the active session with Bob since her Container sent a notification to each of the registered apps

File Transfer Scenario (cont.)

Bob's Container maintains a list of registered applications, including local apps and remote apps, communicating with them about call status and invoking applications on demand



❹ Since Bob often moves between devices, he does not set a default destination for file transfer requests, so his Container prompts for a destination.

❺ Bob selects his computer (at the office), which results in the file transfer request being directed to his computer.

❻ When Bob arrives at the office, the file will be sitting on his computer. Even if he "hangs up the call", the file transfer will continue until completed or forcefully terminated.

“Move” Scenario

❶ Bob is walking down the hall when Alice calls to talk to him about the presentation she is reviewing. He takes the call on his mobile phone as he heads to a telepresence-enabled conference room.

❷ Arriving in the telepresence-enabled room, he places his mobile communicator on the table, at which point NFC in the phone detects the telepresence system and initiates an application “move”. This moves the audio from his phone to the Telepresence system, so he can continue the voice call (which then transforms into audio/video). Meantime, Alice initiates an application sharing session.

Bob’s Container was configured to automatically establish associations with certain “known” applications when encountered, which included this telepresence system. Further, his Container was configured to automatically “move” certain active, foreground applications when the association is made.

Being Connected Anywhere

Imagine being able to move from work, home, or anywhere else and having the applications around you at your disposal at all time, including ones that are permanently registered and ones that you might use temporarily.

Automatic Selection of Application Sets

Alice has three difference Application Sets defined: home, roaming, and office. Her Container is configured to automatically select the active Application Set based on her location, which may be determined through local mobile towers, Wi-Fi SSID, GPS coordinates, etc. All applications may remain registered and available at all times, though only applications in the active set will be offered when incoming requests are made and selected automatically when no conflicts exist. The yellow star indicates the default application(s) to invoke in each profile when initiating a session from the Container.

Home Profile *(active)*
- Voice/Video (home desk phone) ★
- File Transfer (home computer)
- Whiteboard (home computer)
- App Share (home computer)

Roaming Profile
- Voice/Video (mobile handset) ★
- File Transfer (mobile handset)
- File Transfer (office computer)

Office Profile
- Voice/Video (office desk phone) ★
- File Transfer (office computer)
- Whiteboard (office whiteboard)
- App Share (office video display)

Home Profile
- Voice/Video (home desk phone) ★
- File Transfer (home computer)
- Whiteboard (home computer)
- App Share (home computer)

Roaming Profile *(active)*
- Voice/Video (mobile handset) ★
- File Transfer (mobile handset)
- File Transfer (office computer)

Office Profile
- Voice/Video (office desk phone) ★
- File Transfer (office computer)
- Whiteboard (office whiteboard)
- App Share (office video display)

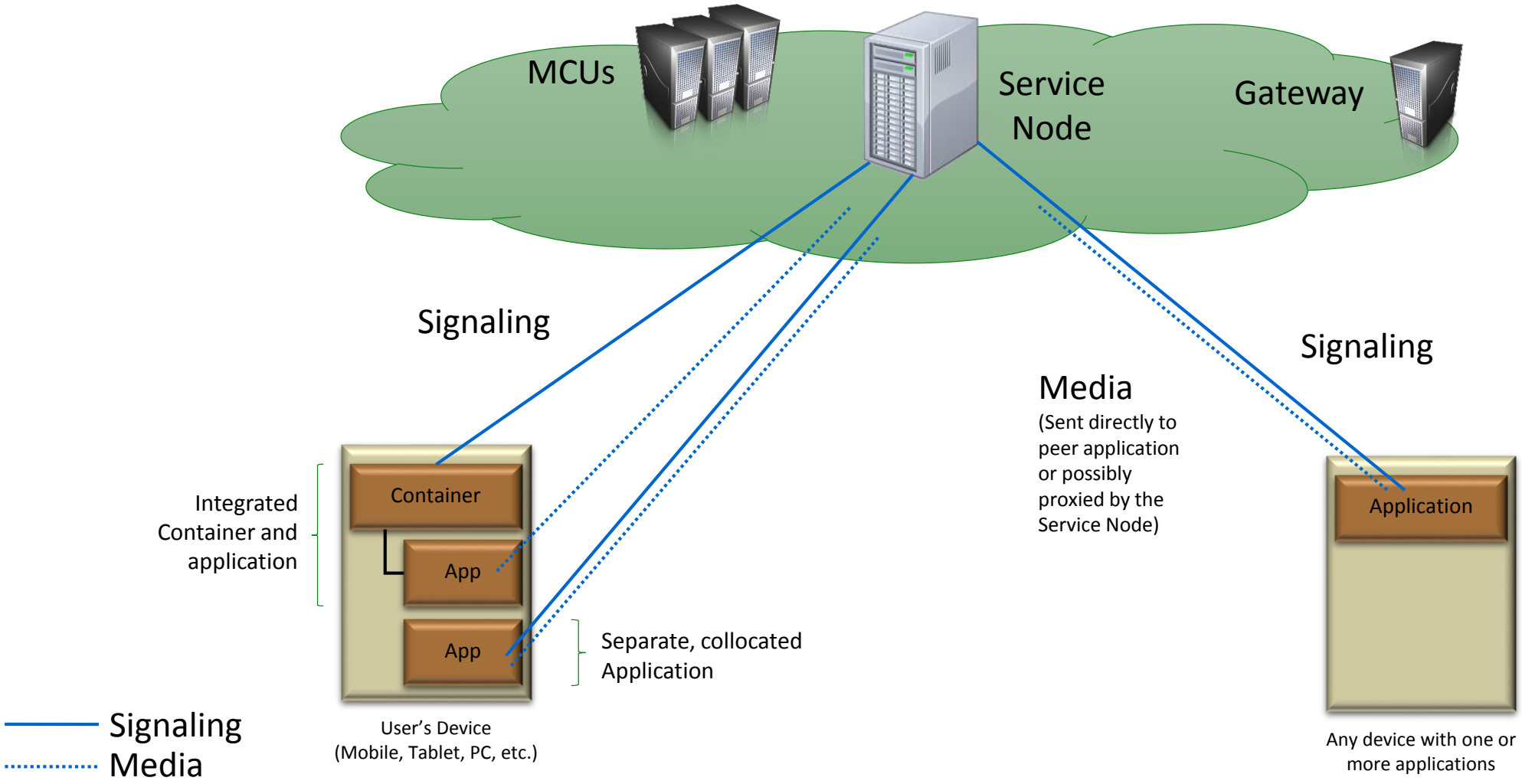
Home Profile
- Voice/Video (home desk phone) ★
- File Transfer (home computer)
- Whiteboard (home computer)
- App Share (home computer)

Roaming Profile
- Voice/Video (mobile handset) ★
- File Transfer (mobile handset)
- File Transfer (office computer)

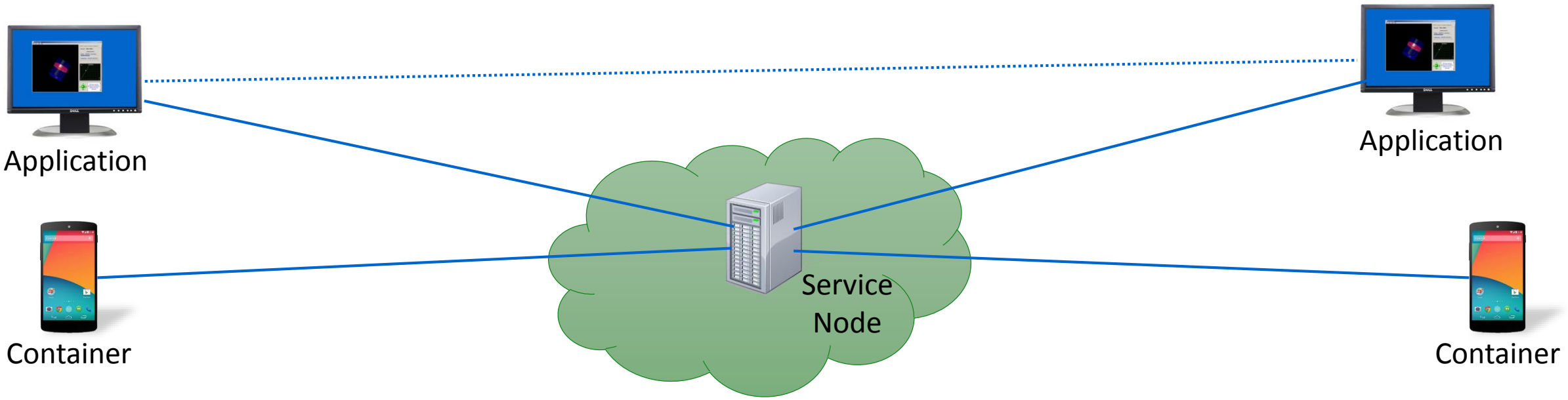
Office Profile *(active)*
- Voice/Video (office desk phone) ★
- File Transfer (office computer)
- Whiteboard (office whiteboard)
- App Share (office video display)

Architecture

AMS High Level Architecture



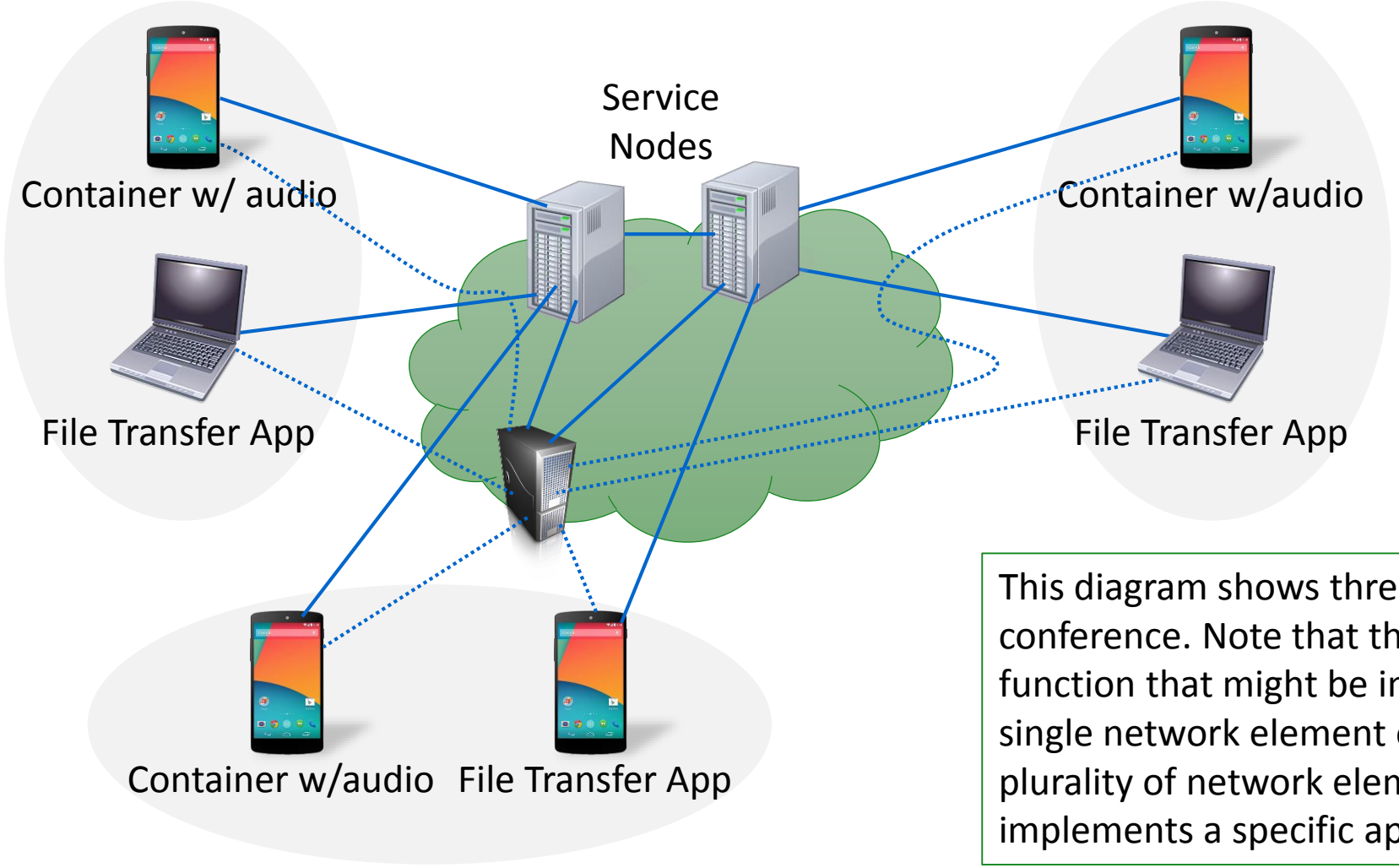
Two Communicating Assemblages



Signaling
Media

Not depicted here are the associations between the Container and its applications. This is implied by physical proximity. The next slide will go through the registration and association processes.

Conferencing Scenario with an MCU

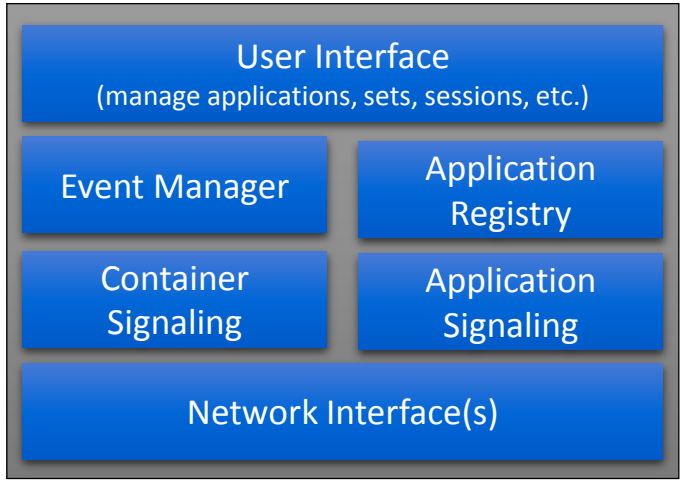


This diagram shows three devices in conference. Note that the MCU is a logical function that might be implemented as a single network element or distributed across a plurality of network elements, each of which implements a specific application.

The Container

- The Container is responsible for providing a user interface, network signaling, and coordinating communication between applications
- The Container is not responsible for actually implementing or even understanding application behavior
- The Container is not a “voice over IP” system, but rather a functional element serving as a communication facilitator

Container

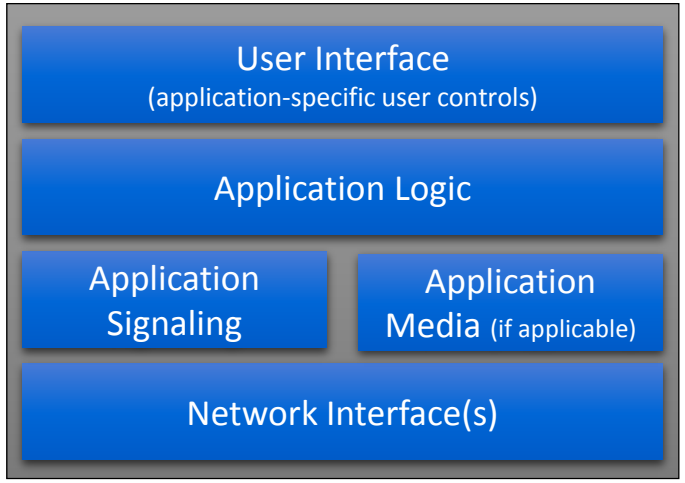


Note: These functions are still being defined and this list is only to illustrate the separation of “container” logic and “application” logic

Applications

- Applications may exist on the same device as the Container or separate devices
- AMS does not impose restrictions on application functionality; developers are free to be creative
- A user may utilize any number of applications at the same time
- Applications associate with a Container
- Applications interact with remote applications
- Applications may be upgraded independently of the Container or other applications
- The ITU will standardize some applications to ensure interoperability
- Non-standard applications, including legacy applications integrated with AMS, will offer value to users, as developers can be creative and introduce a host of applications that improve productivity

Application

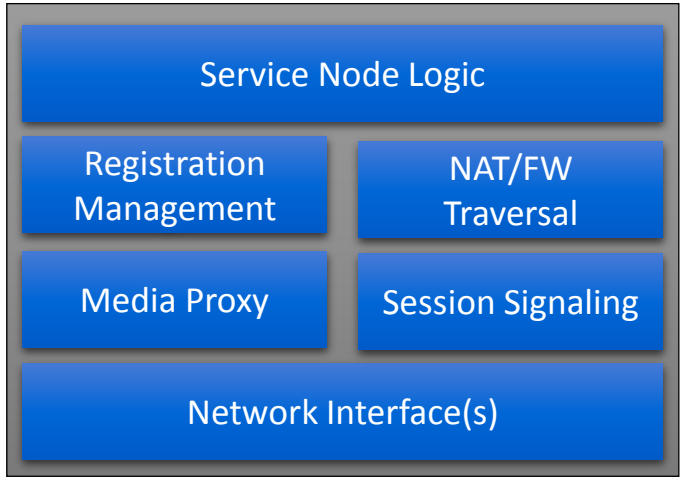


Note: Beyond communicating with the Container and processing certain required messages, most of the application is focused on application functionality that is entirely separate from other entities in the system

Service Node

- The Service Node performs a number of tasks
  - Handles Container and application registrations
  - Facilitates NAT/FW traversal
  - Proxies media
  - Routes signaling messages
- It is expected that some functions, such as media proxy, may be physically separated from the physical device handling registrations
- Applications and the Container are not necessarily registered to the same Service Node or network; messages are routed between nodes transparently to end user devices
- To the extent possible, IP addresses will not be utilized directly by end devices
  - URIs will be used to address each entity uniquely
  - IP addresses will be conveyed in the establishment of media flows

Service Node

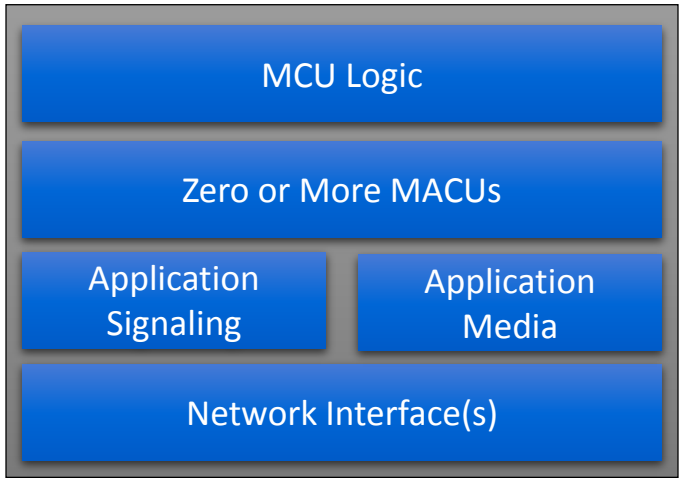


Note: These functions are still being defined and this list is only to illustrate those functions identified. It is possible to distribute these functions across a plurality of physical devices

Multipoint Control Unit (MCU)

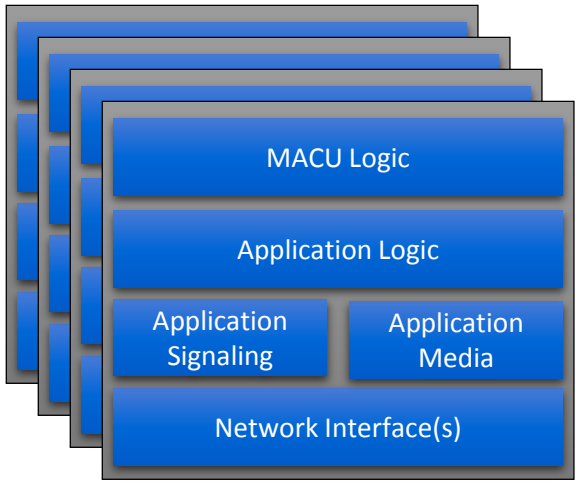
- A Multipoint Control Unit (MCU) will provide multipoint support for AMS
- It is possible for applications that intrinsically understand multipoint to handle multipoint autonomously without the need of the MCU to perform any media processing
  - At present, no conference identifier is defined in AMS, though one can be derived from the SessionID (as the MCU's UUID component will be common for all entities in the conference) or one may be introduced if needed
- An MCU is a logical entity that might be broken into multiple physical devices for scaling, geographic distribution, and media optimization (referred to as "cascading")
- MCU may utilize Multipoint Application Control Units (MACUs) that implement specific applications (e.g., whiteboard or file transfer), either co-located or physically separate
- A conference control application may be associated with the Container to control conference membership
- This breaks away from the legacy monolithic design to one where functionality is introduced incrementally as new applications are developed (time-to-market, limits interdependencies, "plug and play")

Multipoint Control Unit



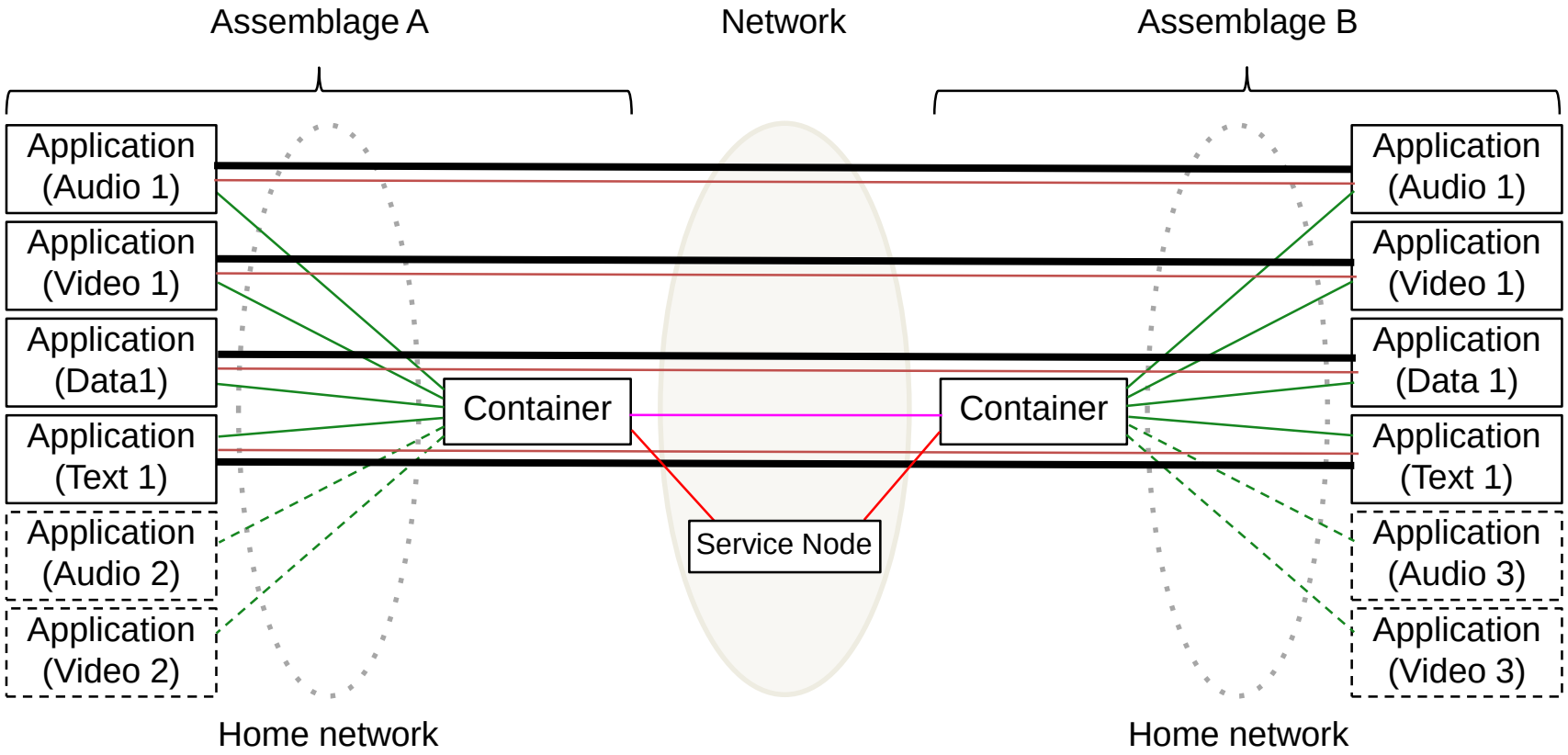
Note: The MACU logic may be physically separated into one or more external entities, where an MACU might implement one or more application protocols and itself be cascaded or logically split between control and media processing.

Multipoint Application Control Unit



Protocol Overview

Several Interfaces Identified



Interface A: Container to Container signaling (usually via the SN)
Interface B: Application to Container signaling (usually via the SN)
Interface C: Media information (may be proxied in the network)
Interface E: Network access (Container/Application to SN) signaling
Interface K: Application to Application signaling (may be via the SN)

This looks more complex than it really is. This will become more obvious on the subsequent slides. These interfaces are formally named to reduce ambiguity in the specifications.

Message Primitives

- The intent is to minimize the number of message primitives, with the current messages under consideration
  - Initiate
  - Engaged
  - Terminate
  - Information
- Treatment of a message depends on the message type and the “action” member in the “body” member

High Level Structure of Signaling Messages

- A messages comprises the following data elements
  - Session-ID – uniquely identifies the end-to-end session
  - Sender – the URI(s) of the sending entity
  - Recipient – the URI(s) of the recipient
  - Sequence – a message sequence numbers
  - AckSequence – used to facilitate message acknowledgement
  - MessageType – One of the basic message primitives (see next slide)
  - Body – the actual body of the message, which varies by message type
    - May be in plaintext or encrypted (the latter for end-to-end encryption)
    - Includes an “action” member that further defines the action to take for the message
    - Includes a “genericData” member for non-standard or basic extensibility
    - Includes a “content” member, the type and value of which depends on “action”

Signaling Syntax and Message Encoding

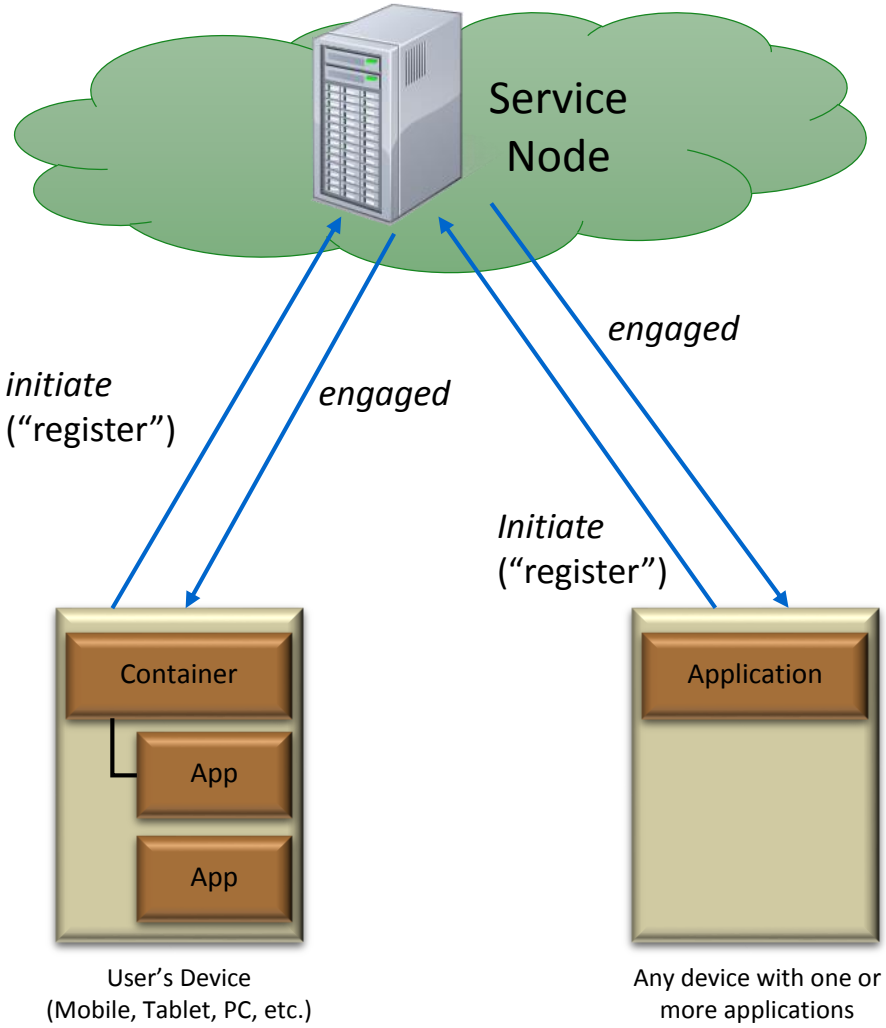
- Signaling syntax will be defined in JSON, with one message per line (i.e., LF or CR will only be present at the end of a message)
- Intent is to use draft-newton-json-content-rules for formal syntax specification (or similar)
- Application-level acknowledgement to address connection breakage
- Same syntax and encoding rules used for registration, associations, and session establishment


```
{
  "sessionID" : [ "7d2d4b8a-4488-11e3-b38c-f4ea67801e29" ],
  "sender" :
  {
    "displayNames" : [ { "name" : "Alice", "language" : "en" } ],
    "aliases" : [ "ams:alice@example.com" ]
  },
  "recipient" :
  {
    "aliases" : [ "ams:service_node@example.com" ]
  },
  "sequence" : 1,
  "messageType" : "initiate",
  "body" :
  {
    "action" : "register",
    "content" :
    {
      "entities" :
      [
        "urn:itu:t:rec:h.325:p:container",
        "urn:itu:t:rec:h.325:p:app"
      ],
      "credentials" :
      {
        "username" : "alice"
      }
    }
  }
}
```


Example of a registration message. Note that this entity is both a Container and an application.

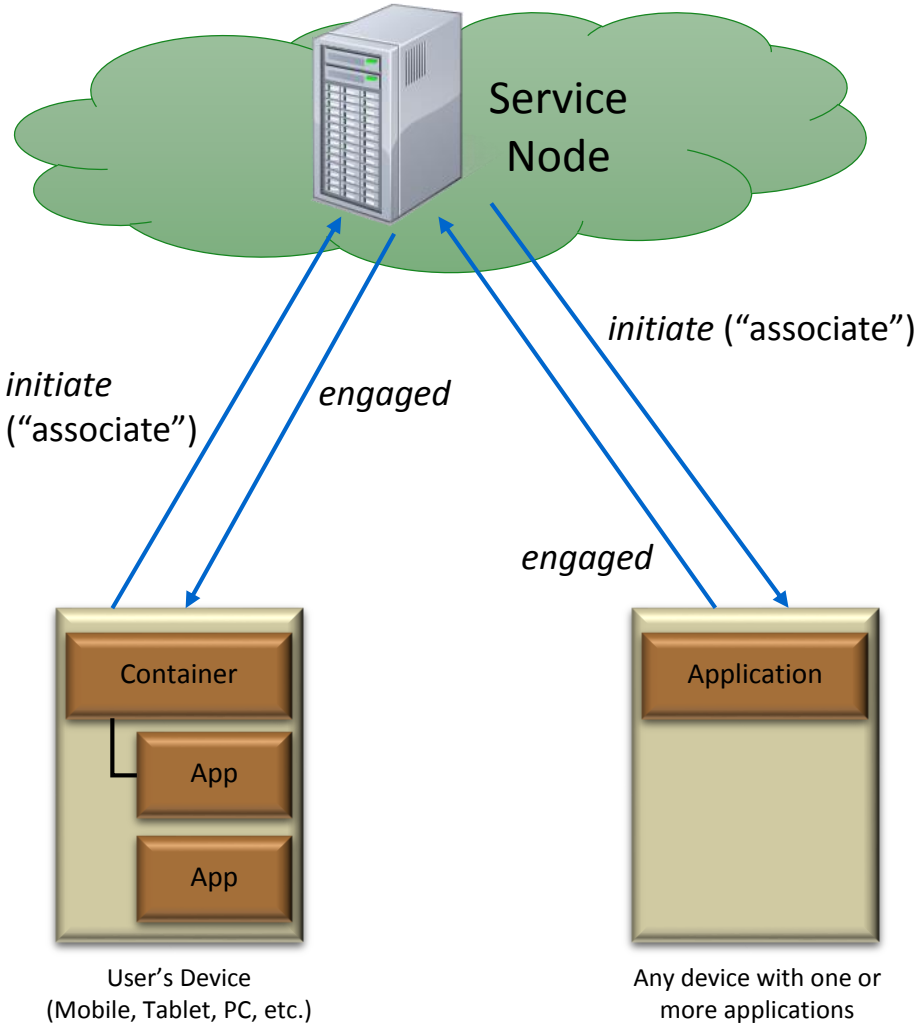
Registering with the Service Node

- Before a Container or application can do anything, it must register with a Service Node
  - Note: technically, it should be possible to communicate directly
- The Service Node may assign any number of alias addresses (e.g., ams:user@example.com and tel:+18005551212)
- Registrations are a type of session that persist until either end terminates the registration
- The signaling transport may be anything, though it must be possible for signaling to be initiated by either end at any time; TLS is an initial candidate
- TLS connections are expected to be persistent and re-established by the Container or application if lost unexpectedly



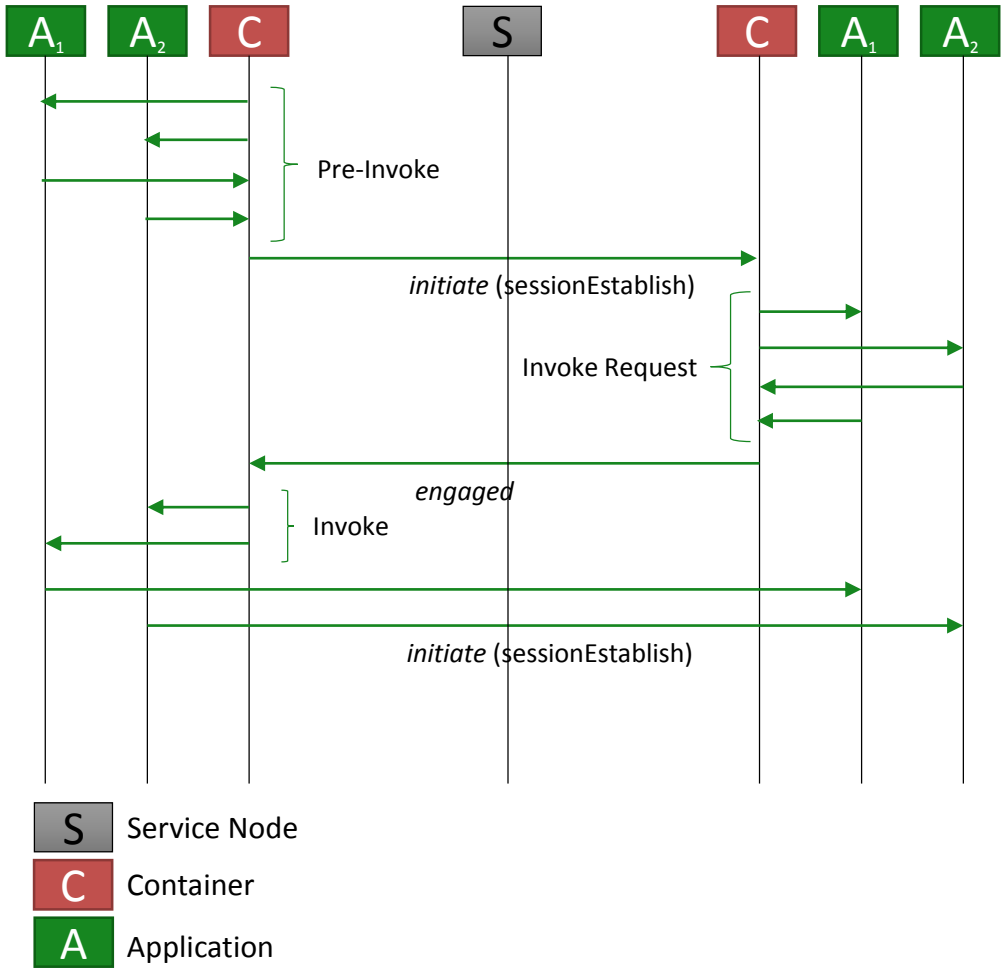
Associating a Container with Applications

- Containers may initiate associations with applications or applications may initiate associations with Containers, the addresses for which may be manually entered, scanned QR code, or discovered using NFC
- Containers shall not associate with other Containers; applications shall not associate with other applications
- Applications provide an association token to the Container
- Associations may persist for a specified time indicated by the responding entity, though associations may be terminated at any time
- Associations are refreshed on a schedule as defined by the application or Container
- Associated applications are presented to the user with user-friendly names and/or icons



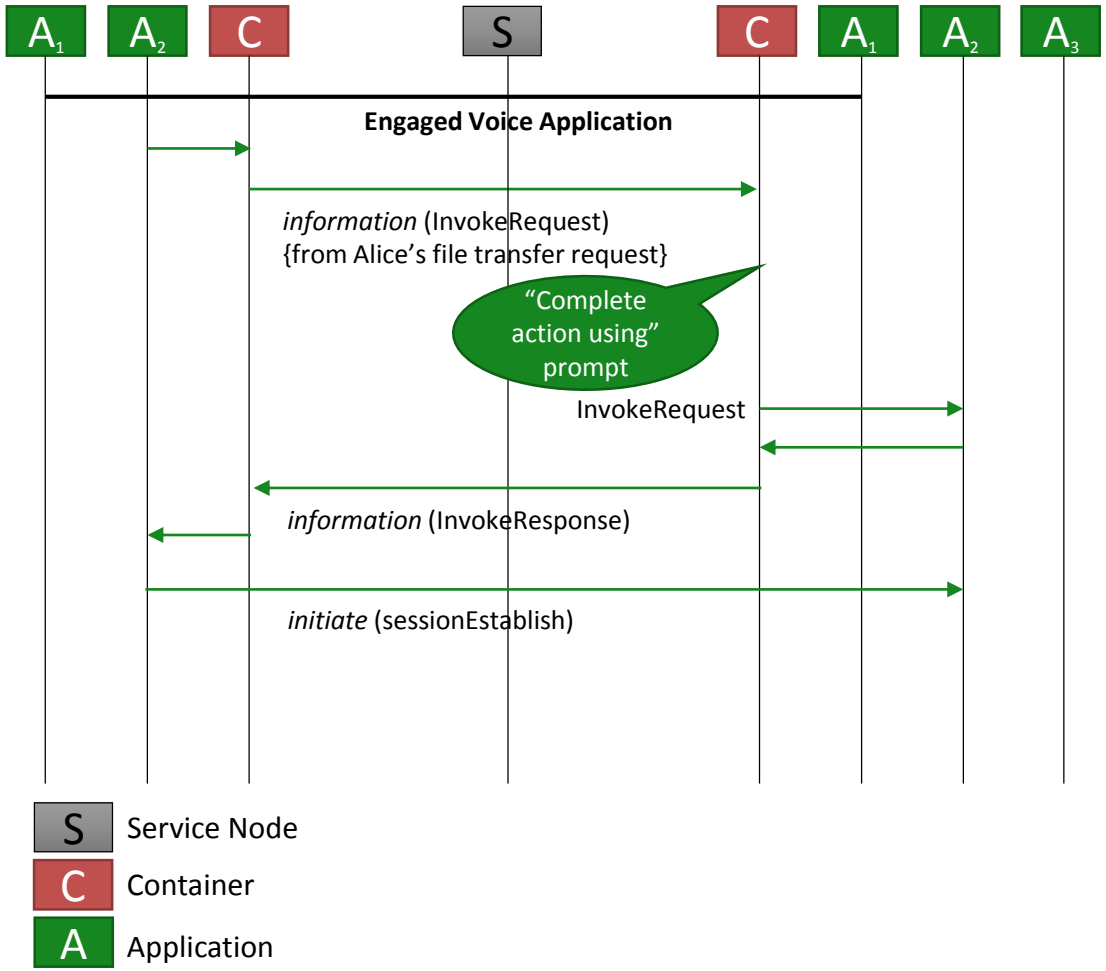
Initiating a Communication Session

- A communication session may be initiated from an application or the Container
- When initiating a new communication session, applications in the current application set are pre-invoked ("Are you ready?")
- The Container establishes communication with the remote Container and requests invocation of applications
- The responding Container will return a set of applications and associated invocation tokens
- The container passes the invocation tokens to pre-invoked applications, matching the application type and tokens
- The applications then initiate communication with the peer application using the provided association token



File Transfer Scenario

- The flow to the right illustrates the signaling that would take place when Alice initiates a file transfer toward Bob, as described earlier in the presentation
- Alice's computers sends an invocation request toward Bob via her Container
- Bob's container (not having a default application) prompts him for a destination, wherein he selects his computer (A_2)
- Bob's Container forwards the request to Bob's computer, which accepts the invocation request
- Bob's Container forwards the response back to Alice's container, which then forwards the response back to her file transfer app
- Alice's file transfer application then initiates a session toward Bob's computer and delivers the file



Packetizer™