

SIEMENS

H.323 and SIP Comparing Their Service Architectures

Dr. Harald Müller

2002-04-09

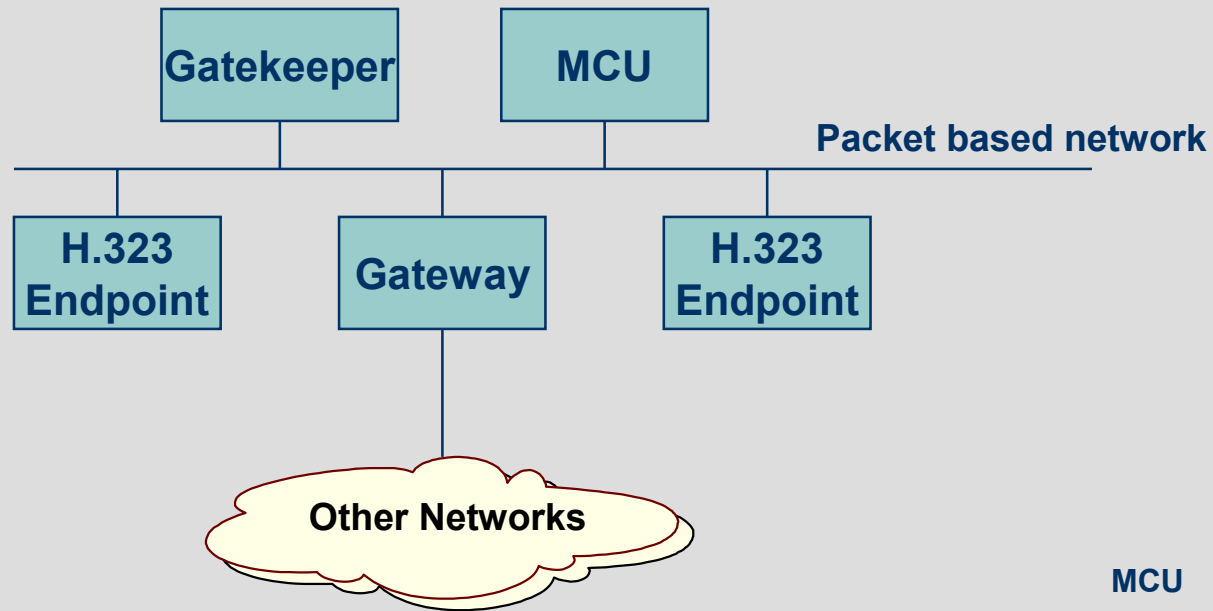
Outline

- Focus
 - Comparison of the two major VoIP protocols H.323 and SIP with respect to their service architectures

- Outline
 - H.323 and SIP Overview
 - Service Implementation Concepts
 - Supplementary Service Example



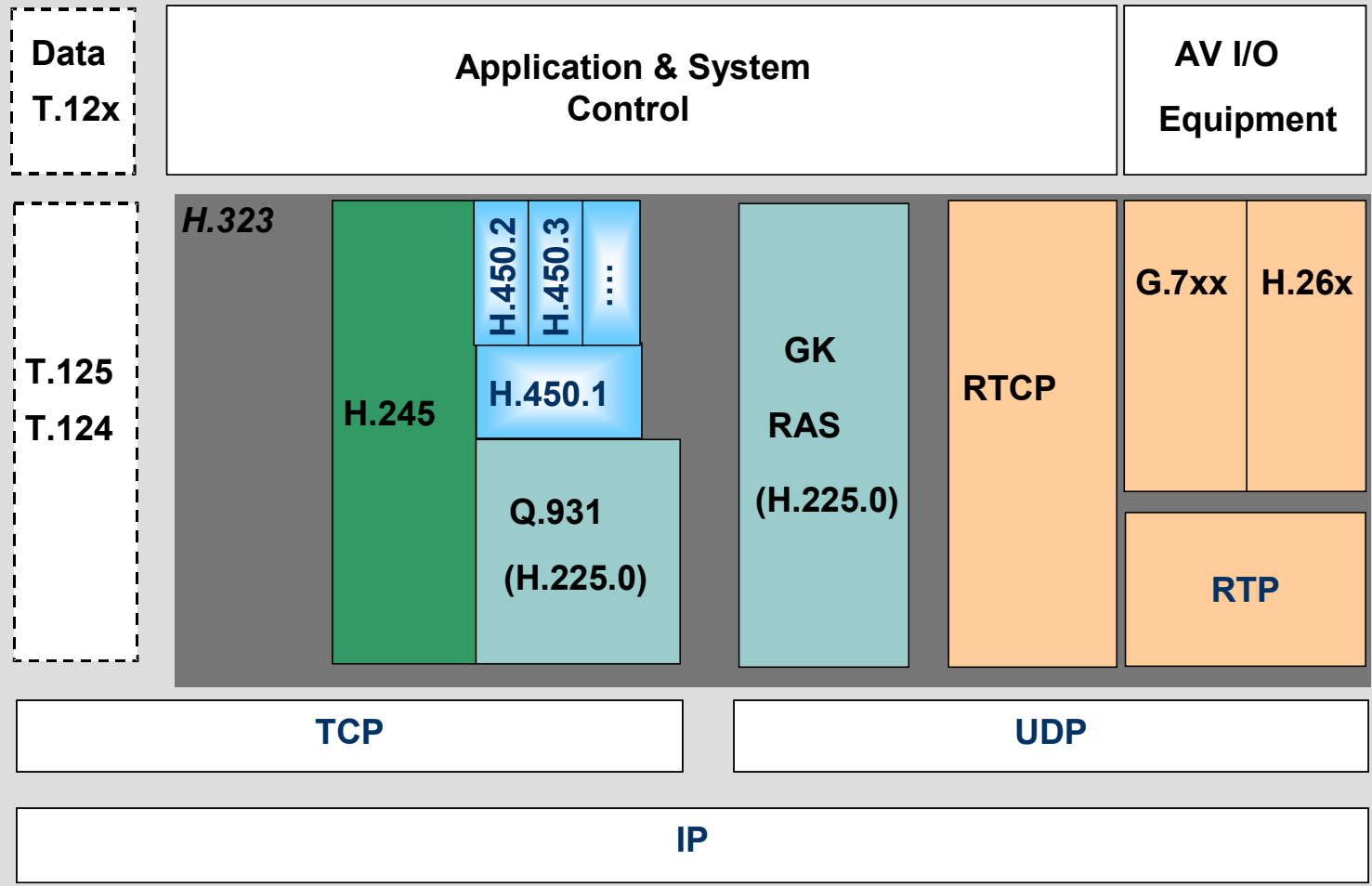
H.323 Architecture



MCU Multipoint Control Unit



H.323 System/Standards



Data
T.12x

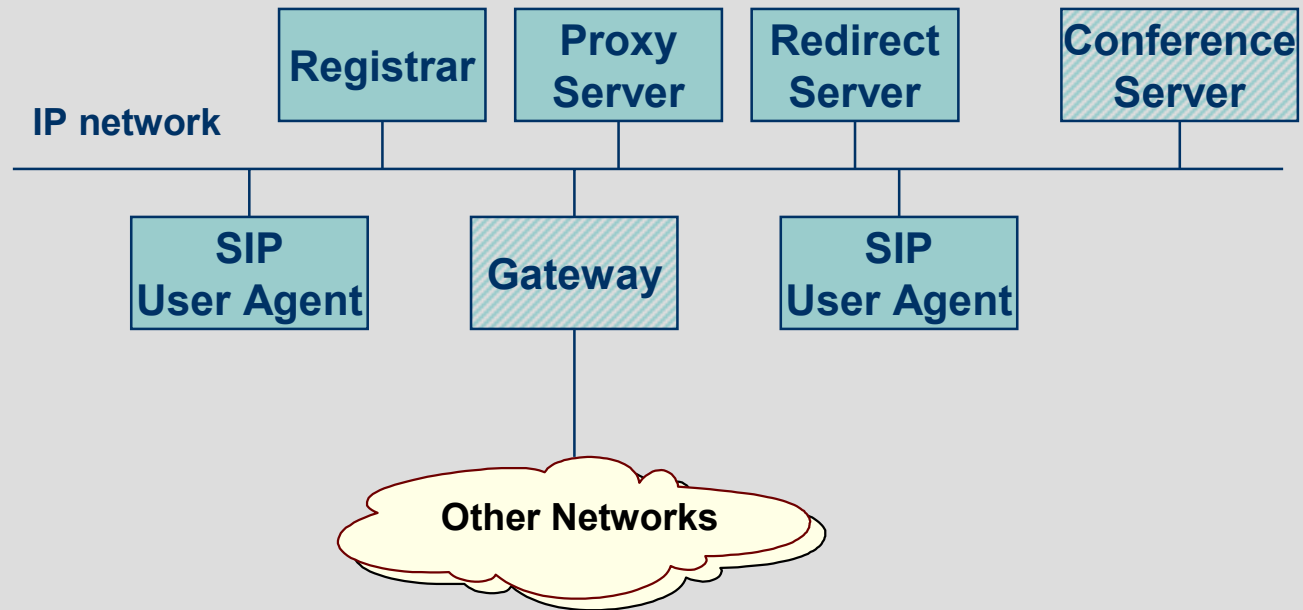
T.125
T.124



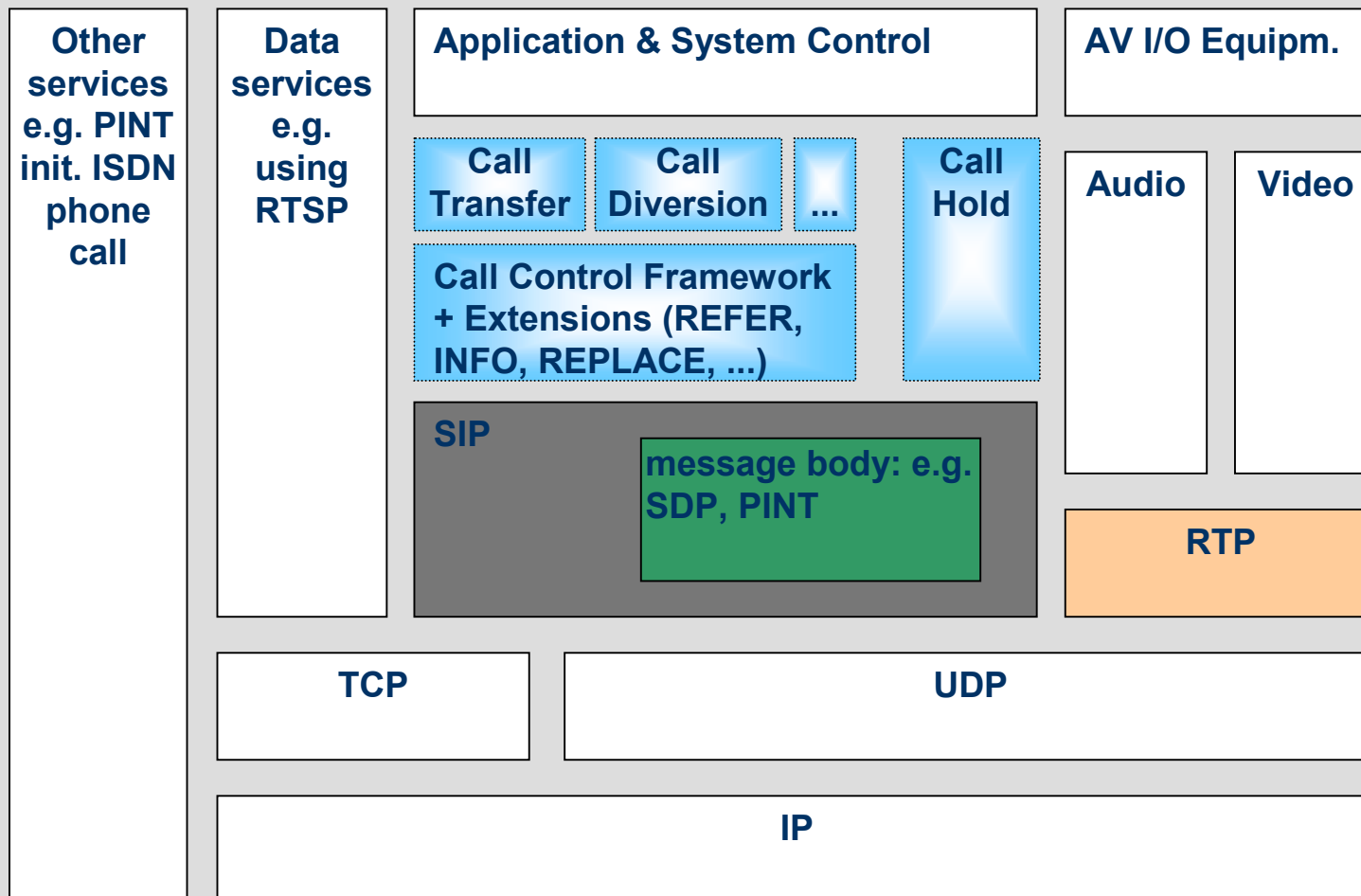
Service Control



SIP Architecture (Example)



SIP System/Standards




Service Control



A First Comparison

SIP and H.323 Architecture Components

	Client	Servers in the Network		
SIP (IETF)	Terminal	Proxy Server, Redirect Server, Registrar	Conference Server*	Gateway*
H.323 (ITU-T)	Terminal	Gatekeeper	MCU	Gateway

*not standardized up to now

SIP and H.323 Protocol Components

	Realtime Data Transmission	Call Control	Feature Control		Signaling Procedure Variants	
			Framework	Extensions		
SIP (IETF)	RTP/RTCP	SIP (SDP)	call control framework*	Transfer*, ...	-	SIP-INVITE Transaction
H.323 (ITU-T)	RTP/RTCP	H.225.0, H.245	H.450.1	H.450.2 – H.450.12	Basic Call Setup	Fast Connect

*Draft



H.323 Service Implementation - Overview

- H.450 Distributed Features
- Feature Combinations
- Stimulus Feature Control
- Application Layer Feature Control



H.323 Service Implementation (1)

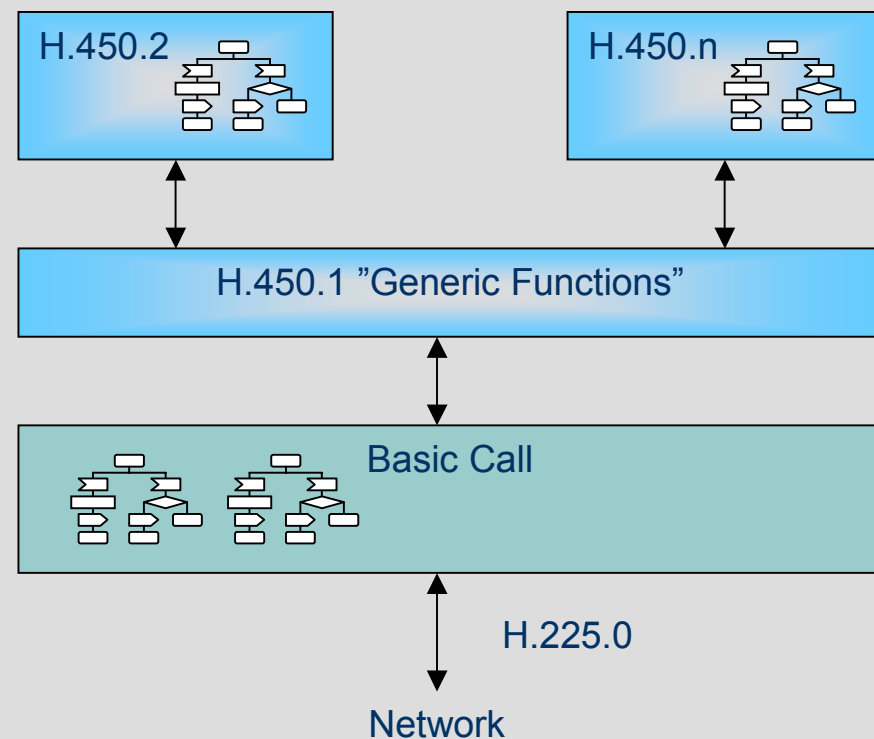
- H.450 Distributed Features
 - 12 Standards with about 25 features available today
 - Framework for dealing with feature complexity and interactions

H.450.1	Generic Functions for Supplementary Services
H.450.2	Call Transfer
H.450.3	Call Diversion
H.450.4	Call Hold
H.450.5	Call Park and Call Pickup
H.450.6	Call Waiting
H.450.7	Message Waiting Indication
H.450.8	Names Identification
H.450.9	Call Completion
H.450.10	Call Offering
H.450.11	Call Intrusion
H.450.12	Common Information



H.323 Service Implementation (2)

- H.450 Framework
 - Explicit and independent state machine for each feature
 - Extension mechanism
 - Generic feature interaction mechanism possible
 - Generic processing of H.450 operations
 - Service negotiation (H.450.12)



H.323 Service Implementation (3)

- Feature Combinations
 - H.450 design philosophy is to define building blocks
 - Feature control above H.450 (client and server) using API (e.g. JTAPI)
 - 1st Party Application Example:
 - ◆ Attendant console =
Basic Call (multiple line) +
Call Hold (H.450.4) +
Call Transfer (H.450.2)
 - 3rd Party Application Example:
 - ◆ ACD (Automatic Call Distribution) =
Basic Call +
Call Transfer (H.450.2) *to music/media server* +
ACD Agent Monitoring using CTI Interface +
CTI-Initiated Call Transfer (H.450.2)



H.323 Service Implementation (4)

- Stimulus Feature Control
 - H.323 Annex L
 - Centralized feature control (stimulus) on top of H.323 basic call (functional)
 - Phone feature package and signaling identical to H.248
 - ◆ Phone capabilities
 - ◆ Key presses
 - ◆ Display messages
 - ◆ Feature Indicators (LEDs)
- Application Layer Feature Control
 - H.323 Annex K
 - Introduces service plane above call control plane
 - Call-related and call independent service control
 - Service control session established by exchanging e.g. URL



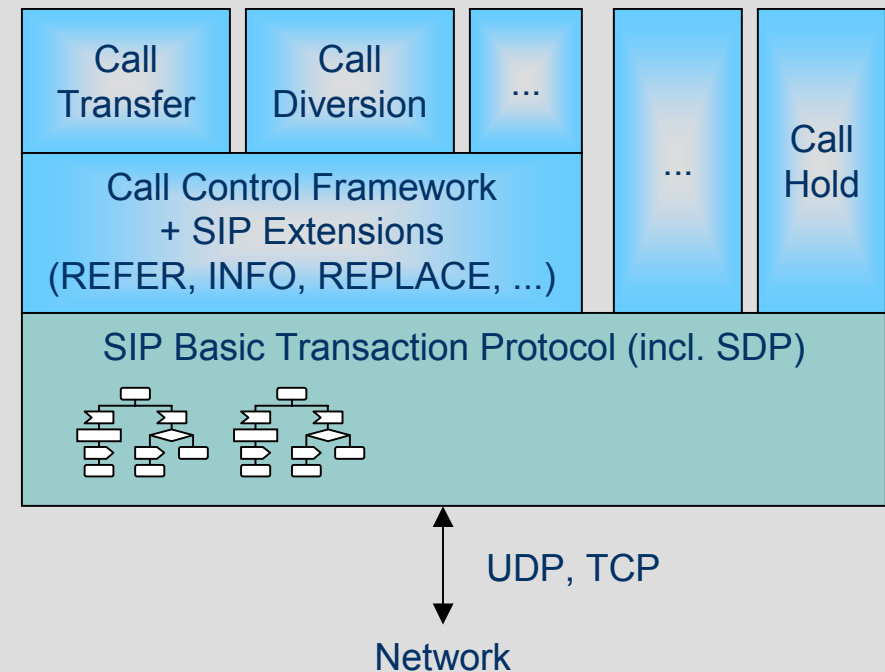
SIP Service Implementation - Overview

- Existing Supplementary Services (RFCs and Drafts)
- Service Negotiation
- Service Programming Languages
 - CPL
 - SIP-CGI
 - SIP Servlets



SIP Service Implementation (1)

- Existing Services (RFCs)
 - Call Hold (non-explicit in SIP basic RFC 2543)
 - Call Transfer (draft)
 - Call Diversion (extensions in framework)
 - Some other drafts with call flows, no RFCs
- Call Control Framework
 - Methodology to define new generic messages and headers and use them for Suppl. services



SIP Service Implementation (2)

- Service Negotiation
 - Simplify the use of call control extensions
 - Headers defined in SIP
 - ◆ REQUIRE
 - Tell the other side (user agent server) which options are expected for the current call
 - ◆ ALLOW
 - Tell the other side (user agent client) which methods are supported by the resource identified by Request-URI
 - ◆ SUPPORTED
 - Enumerates all capabilities of user agent client or server in requests and responses



SIP Service Implementation (3)

- Service Programming Languages
 - On top of SIP basic call flows
 - Three different methods
 - ◆ CPL (Call Processing Language, RFC 2824)
 - ◆ SIP-CGI (Common Gateway Interface, RFC 3050)
 - ◆ SIP Servlets
 - Service logic location
 - ◆ SIP Server (CPL, SIP-CGI)
 - ◆ SIP server or User agent (SIP Servlets)
 - Security
 - ◆ Untrusted users (CPL)
 - ◆ Trusted users (SIP-CGI, SIP Servlets)



SIP Service Implementations - CPL

- Script language for untrusted users
- Users can generate CPL scripts and upload them to a server
- CPL script is processed on the server
- Restrictive function set to avoid security and performance problems
- CPL is independent of the signaling protocol
 - Also suggested for H.323



SIP Service Implementation – SIP-CGI

- Programming solution for trusted users (e.g. server administrators)
- Almost unlimited function set
 - Language independent (e.g. Perl, C, VisualBasic, TCL, ...)
 - Access to all SIP headers (read and write)
 - Most unrestricted service programming language
- Differences to HTTP-CGI
 - Persistence model (stateful processing)
 - Proxying requests (forking)

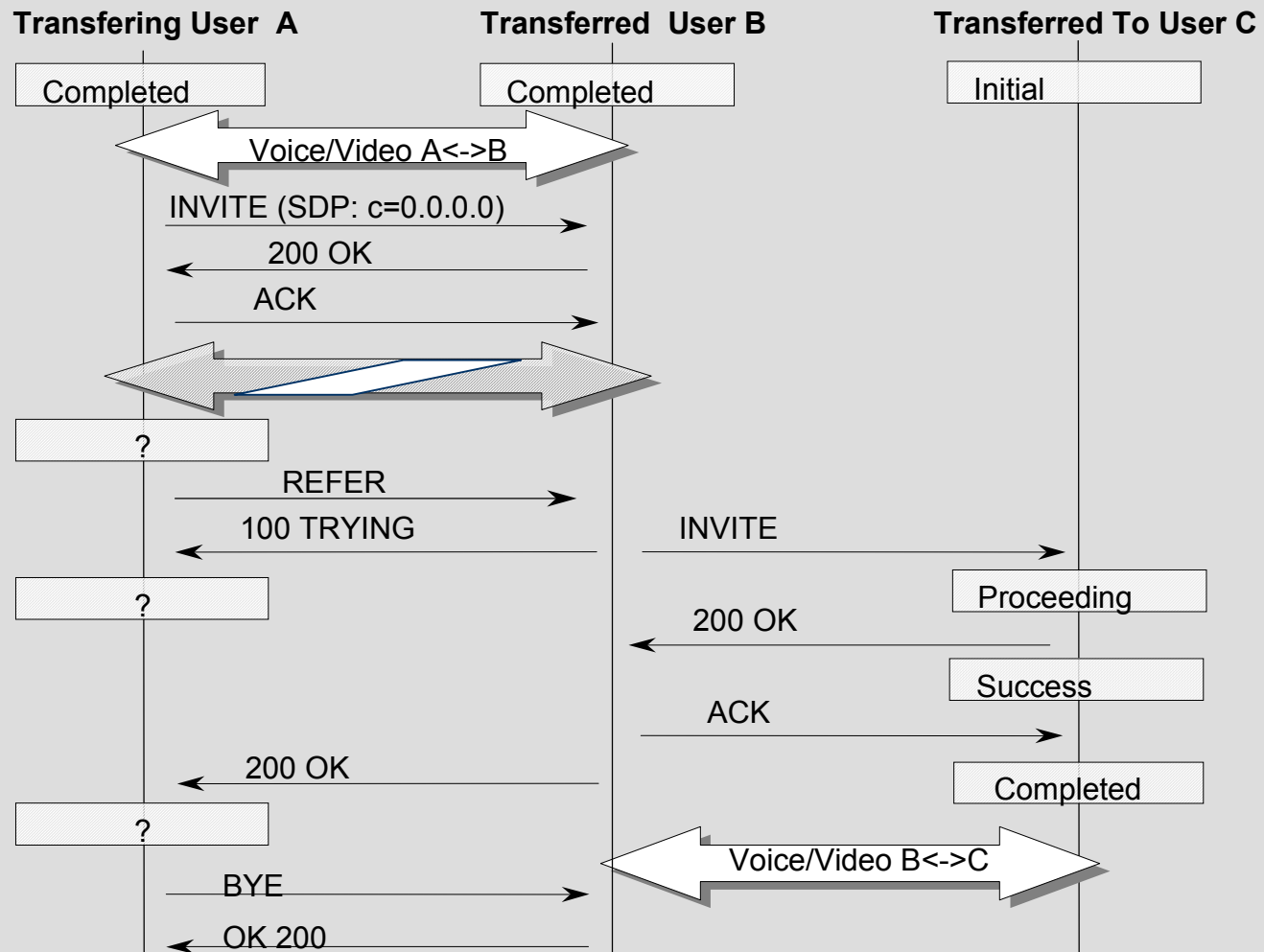


SIP Service Implementation – SIP Servlets

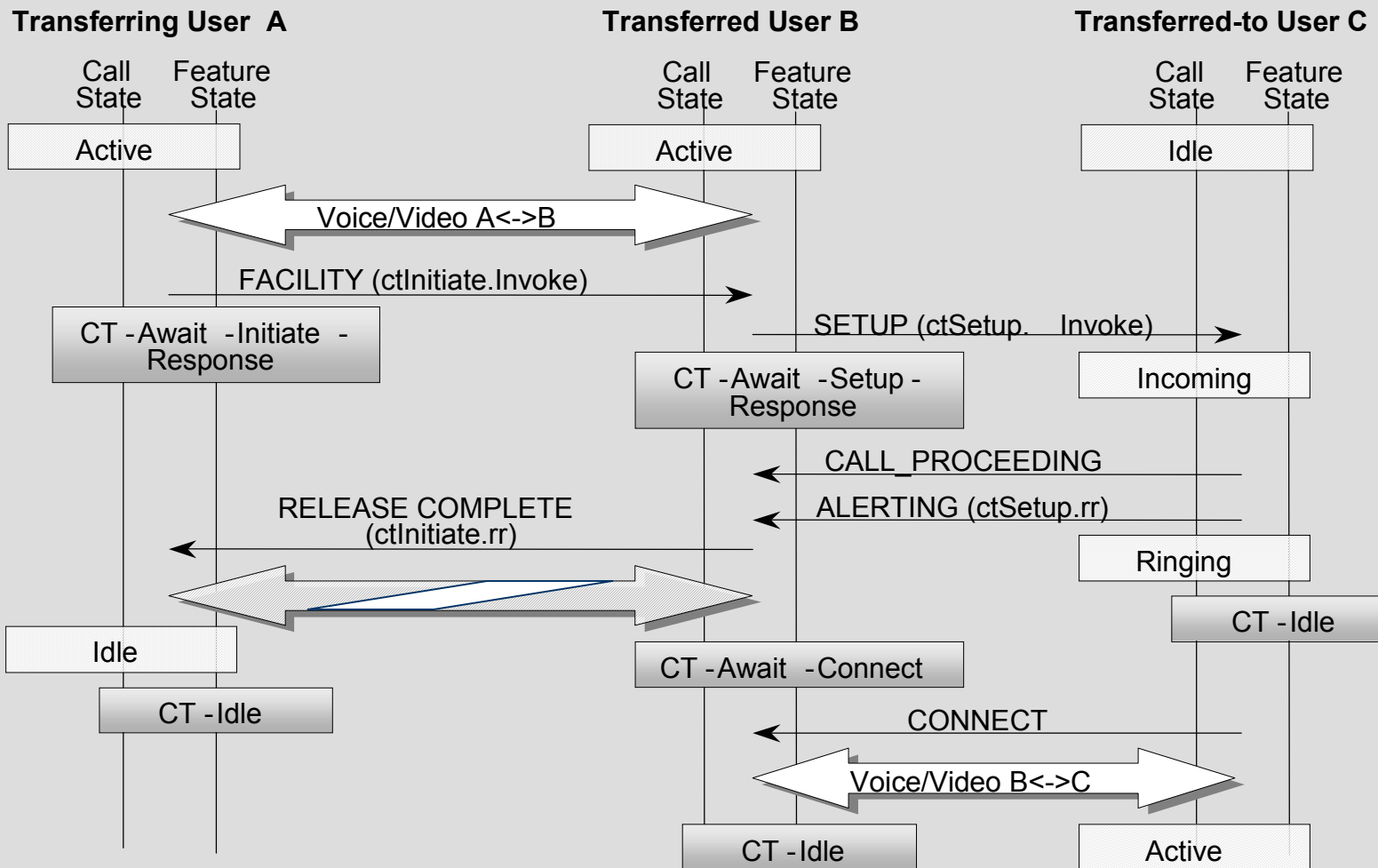
- Compromise between security and function set
- Java API for SIP servers
 - Servlet written in Java
 - Access to all parts of SIP message
- SIP servlet interacts with the servlet engine of the SIP server
 - Servlet engine and Java environment required
 - Runtime verification of servlet commands



Example: Single Step Transfer in SIP



Example: Single Step Transfer in H.323



Comparison from a Service Perspective

- Characteristics of H.323
 - „Tight“ standard
 - Interoperability in open environments
 - Interworking with PSTN and PBX
 - Standard is more mature
 - H.450 framework supports complexity of supplementary services (incl. feature interactions)
 - Rich feature set, still growing
 - Feature set extensible
- Characteristics of SIP
 - „Open“ standard
 - Broader scope for all kinds of applications
 - More flexible in closed environments (all endpoints/servers under single control)
 - Protocol has lower complexity -> better for lightweight non VoIP clients
 - Feature set extensible
 - Several service programming languages described



References and Further Reading

- Full Paper Version
 - Josef Glasmann, Wolfgang Kellerer, Dr. Harald Müller: Service Architectures in H.323 and SIP - A Comparison.
 - ◆ *Whitepaper at Sipcenter,*
http://www.sipcenter.com/files/Service_Architecures_SIP-H323.pdf
 - ◆ *Whitepaper at Packetizer,*
http://packetizer.com/iptel/h323_vs_sip/complis.html
- Other Whitepapers
 - Other SIP/H.323 related Whitepapers
 - ◆ <http://www.sipcenter.com/files/>
 - ◆ http://packetizer.com/iptel/h323_vs_sip/complis.html