Achitecture and Signaling

Multimedia in Packet Networks

H.323 & SIP

upf. Universitat Pompeu Fabra Barcelona

Multimedia in Packet Networks

H.323

SIP

Multimedia in Packet Networks

XLite by CounterPath

Multimedia in Packet Networks

H.323

SIP

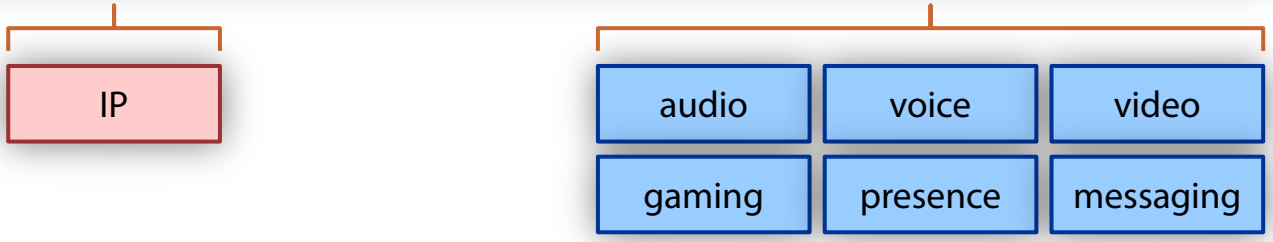
P2PSIP

Multimedia in Packet Networks

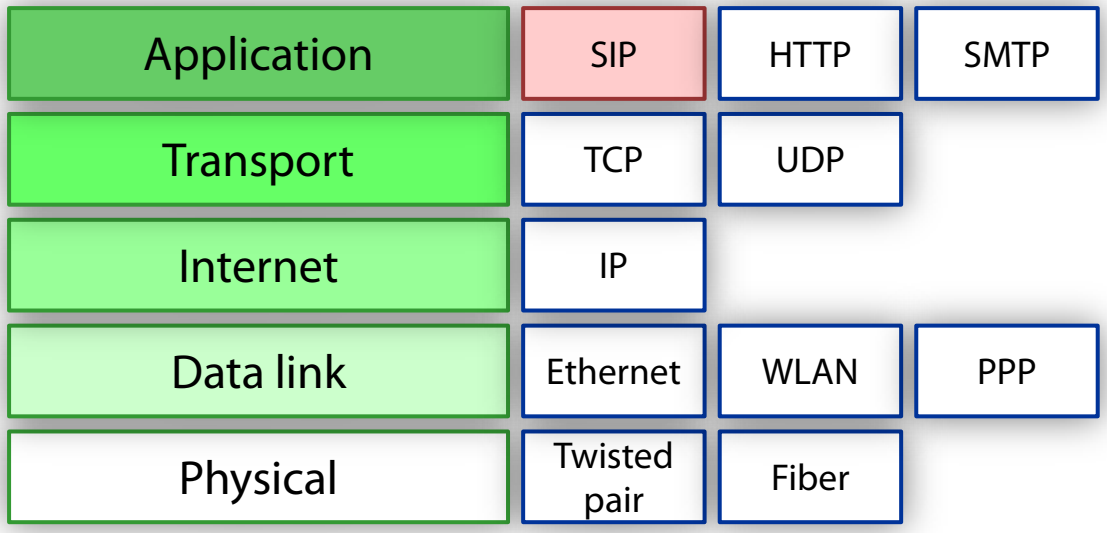
SIP

Session Initiation Protocol

SIP A protocol used to establish, modify and terminate multimedia sessions in the Internet.



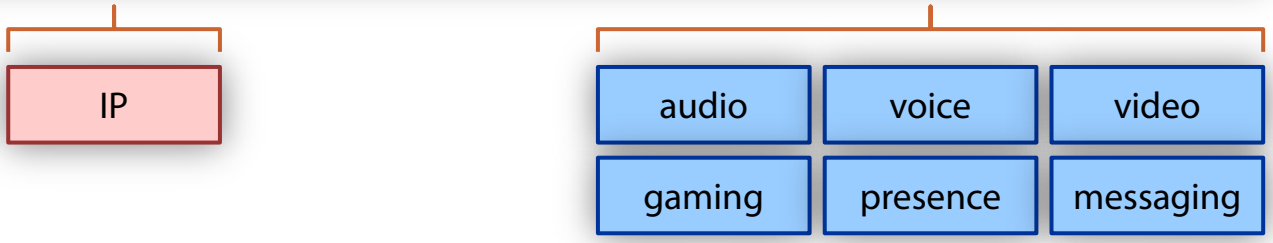
- Key features of SIP
  1. Application Level



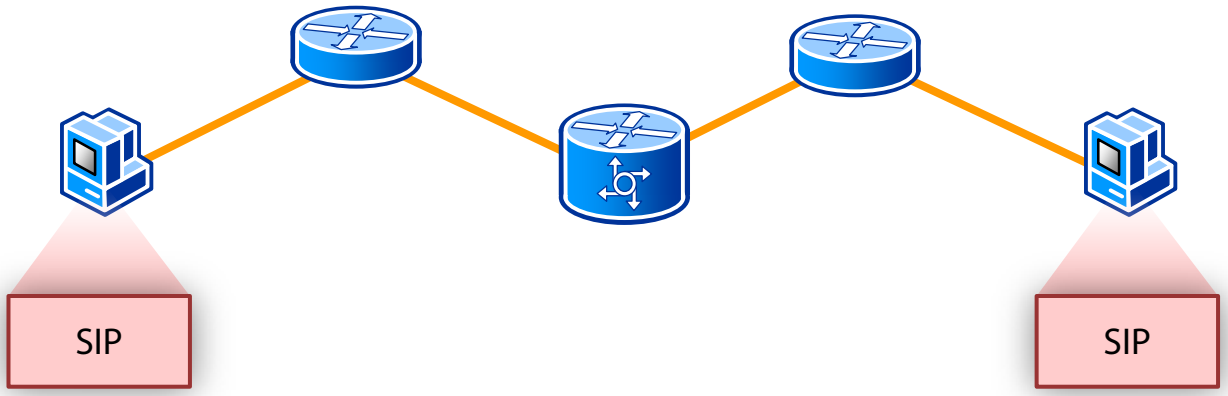
Internet Protocol Layers

Session Initiation Protocol

SIP A protocol used to establish, modify and terminate multimedia sessions in the Internet.

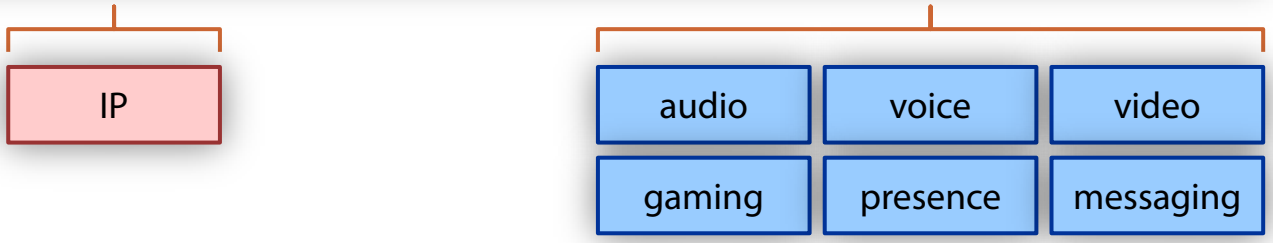


- Key features of SIP
  1. Application Level
  2. End-to-End



Session Initiation Protocol

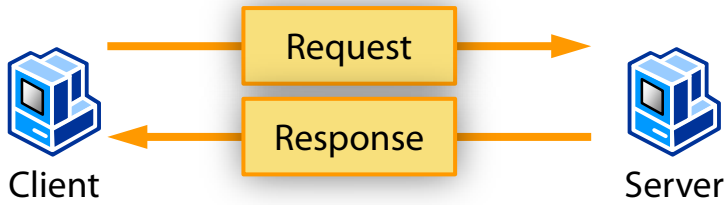
SIP A protocol used to establish, modify and terminate multimedia sessions in the Internet.



- Key features of SIP

1. Application Level
2. End-to-End
3. Client-Server
4. Extensible
5. Text-Based

When exchanging SIP messages



- Easy to add new features to the protocol
- Design based on HTTP and SMTP

History and Standards

1996
- Session Invitation Protocol by M. Handley and E. Schooler
- Simple Conference Invitation Protocol by H. Schulzrinne
- Merged into Session Initiation Protocol (SIP)

1997
- Split SIP into base specification and extensions

1999
- First SIP standard RFC 2543

2002
- Current SIP standard RFC 3261, obsoletes RFC 2543

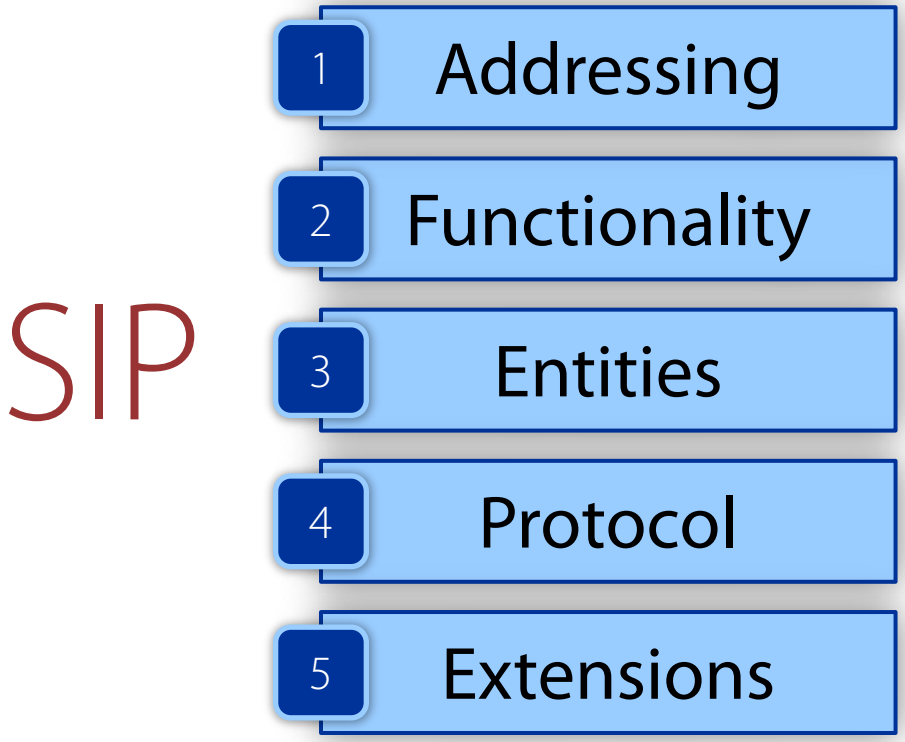
Base standard

RFC 3261

Extensions

RFC 3262	RFC 3263	RFC 3264	RFC 3265
RFC 3853	RFC 4320	RFC 4916	RFC 5393
RFC 5621	RFC 5626	RFC 5630	RFC 5922
RFC 5954	RFC 6141	RFC 6265	

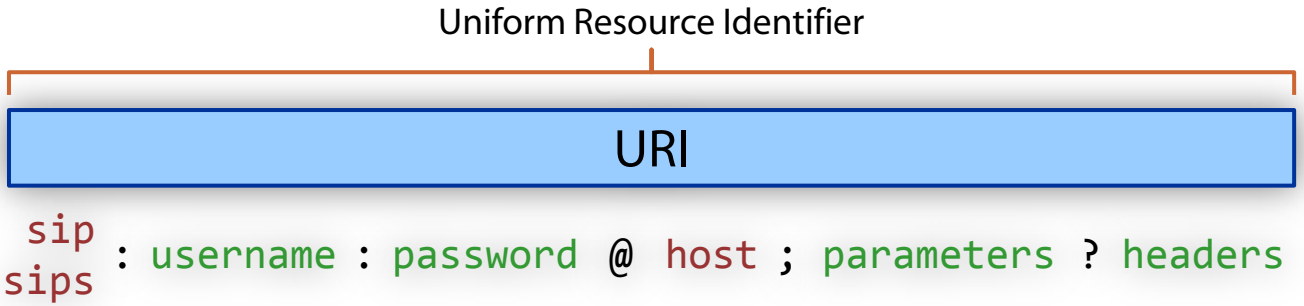
What Do We Study?



Addressing

SIP uses Uniform Resource Identifiers (URI)

They are like web addresses



Examples

```
sip:john.doe@example.com
sips:john.doe@example.com
sip:server.example.com:5060
sip:server.example.com:5060
sip:host.example.com;tag=100
```

Functionality

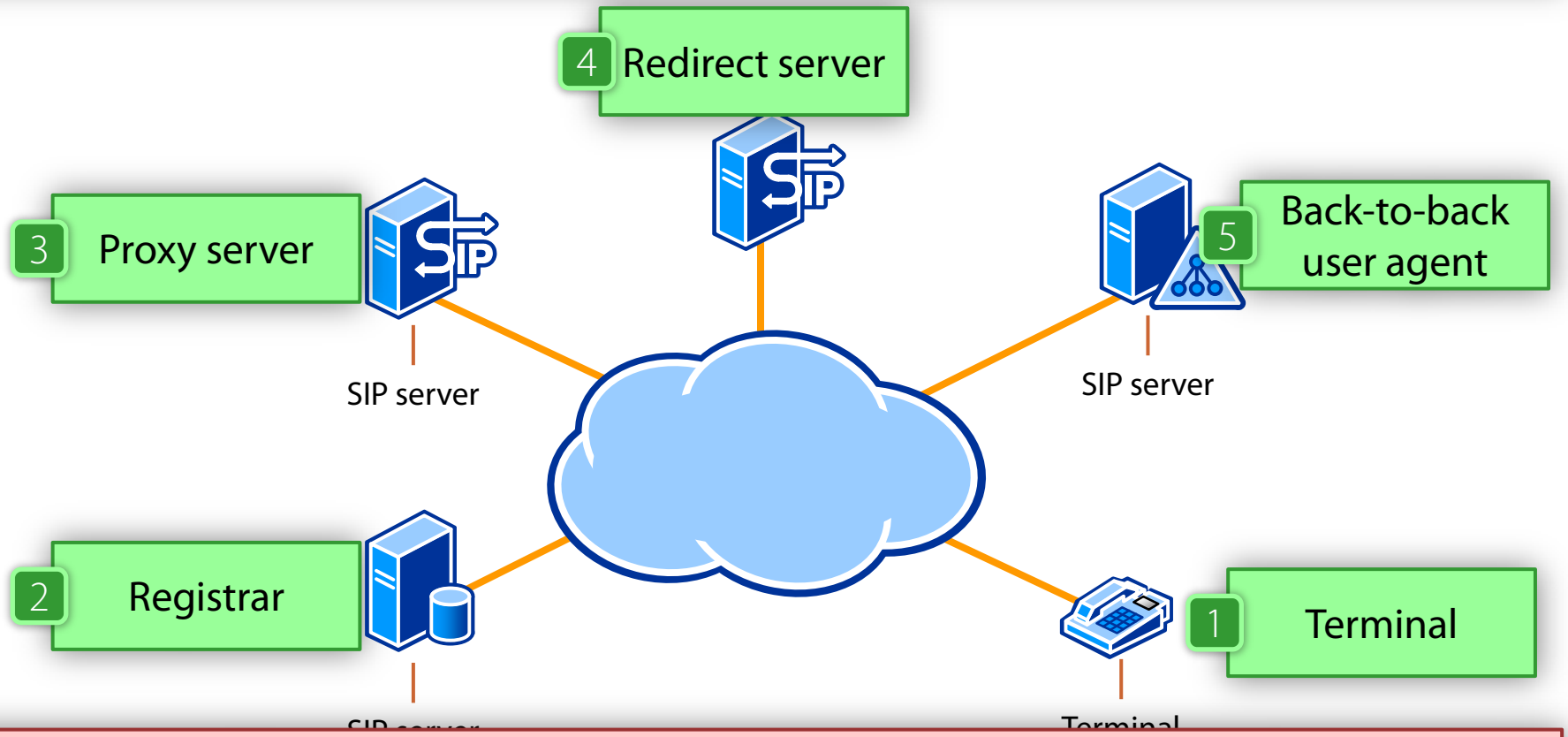
- What does SIP do?

1 User location	• Translates the destination URI to a network (IP) address
2 User availability	• Determines the willingness of the other party to communicate
3 User capabilities	• Determines the network parameters and media types supported by the communicating parties
4 Session setup	• Establishes a session similar to a phone call
5 Session management	• Session transfer, modification and termination

SIP does not specify multimedia services (audio, video)
It allows us to create multimedia services.

SIP Entities

These entities are logical, implemented in software.

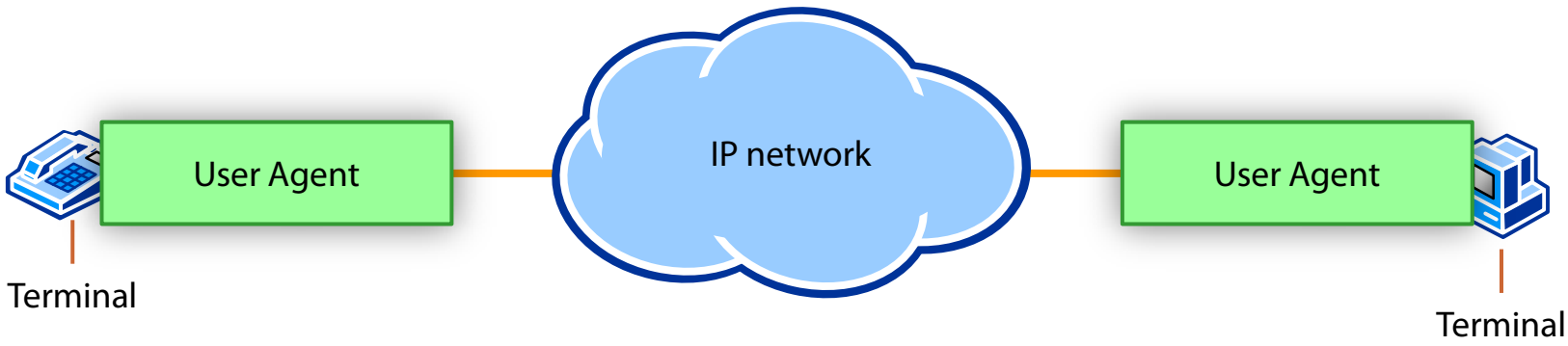


SIP entities consist of one or more user agents (UA)

What Is an User Agent?

The SIP software that processes SIP messages at an endpoint: terminal or server.

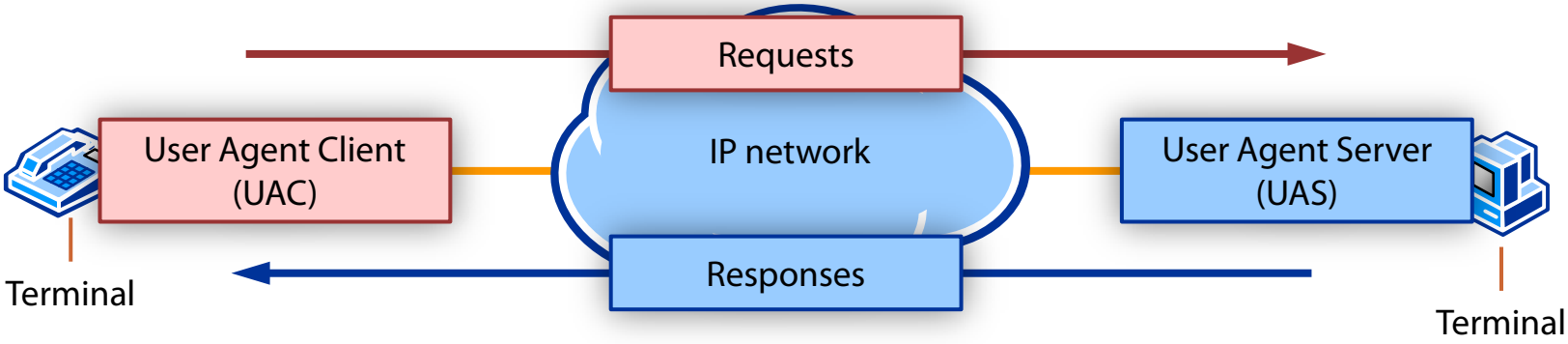
- Let's consider a pair of SIP terminals



What Is an User Agent?

The SIP software that processes SIP messages at an endpoint: terminal or server.

- Let's consider a pair of SIP terminals

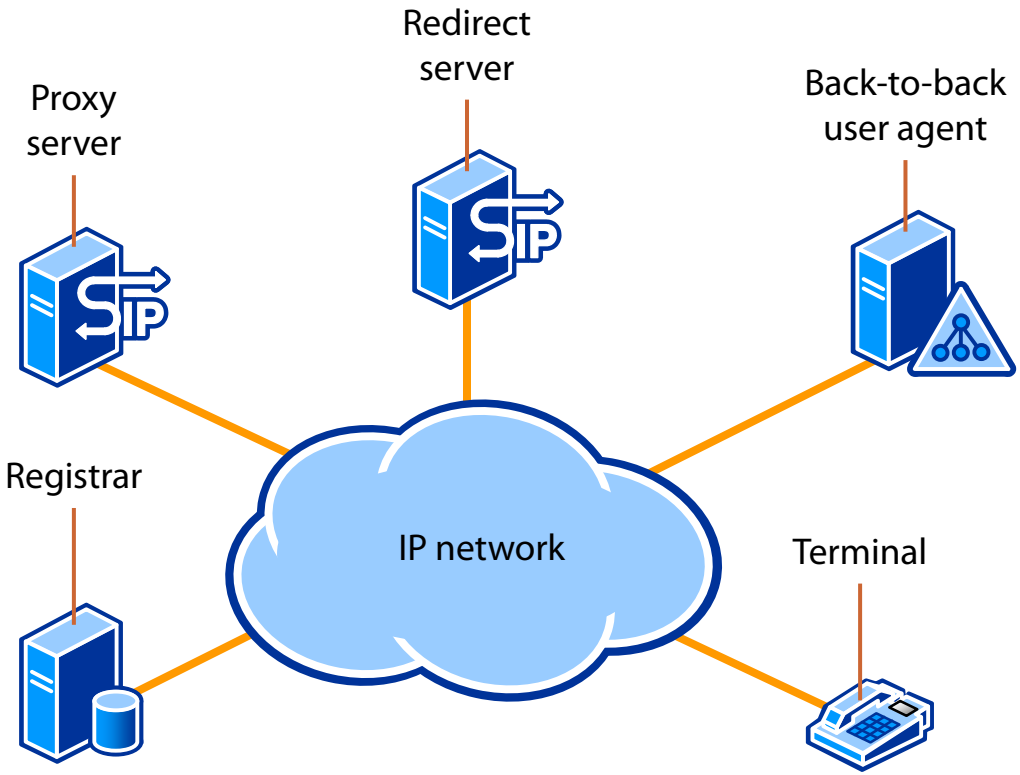


- Initiates sessions (places calls)
- Sends request messages

- Waits for incoming sessions
- Sends response messages

A terminal would implement both a UAC and UAS

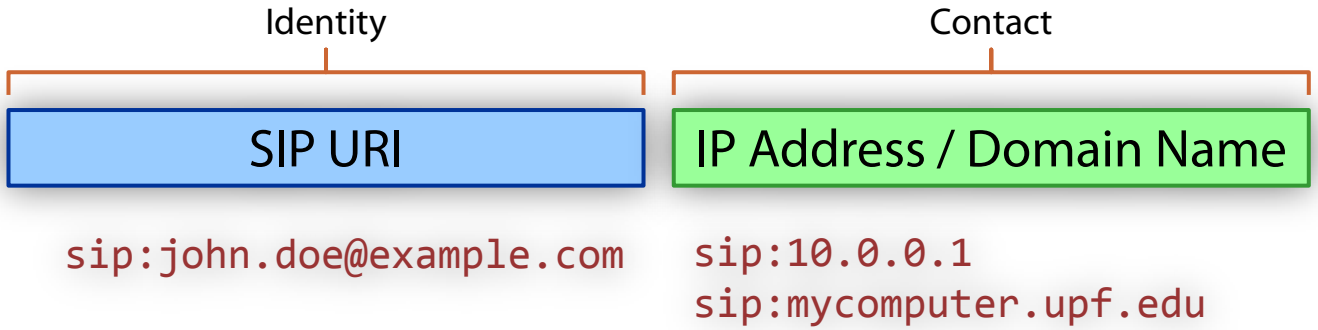
SIP Entities



The Registrar

Is an UAS that accepts only registrations.

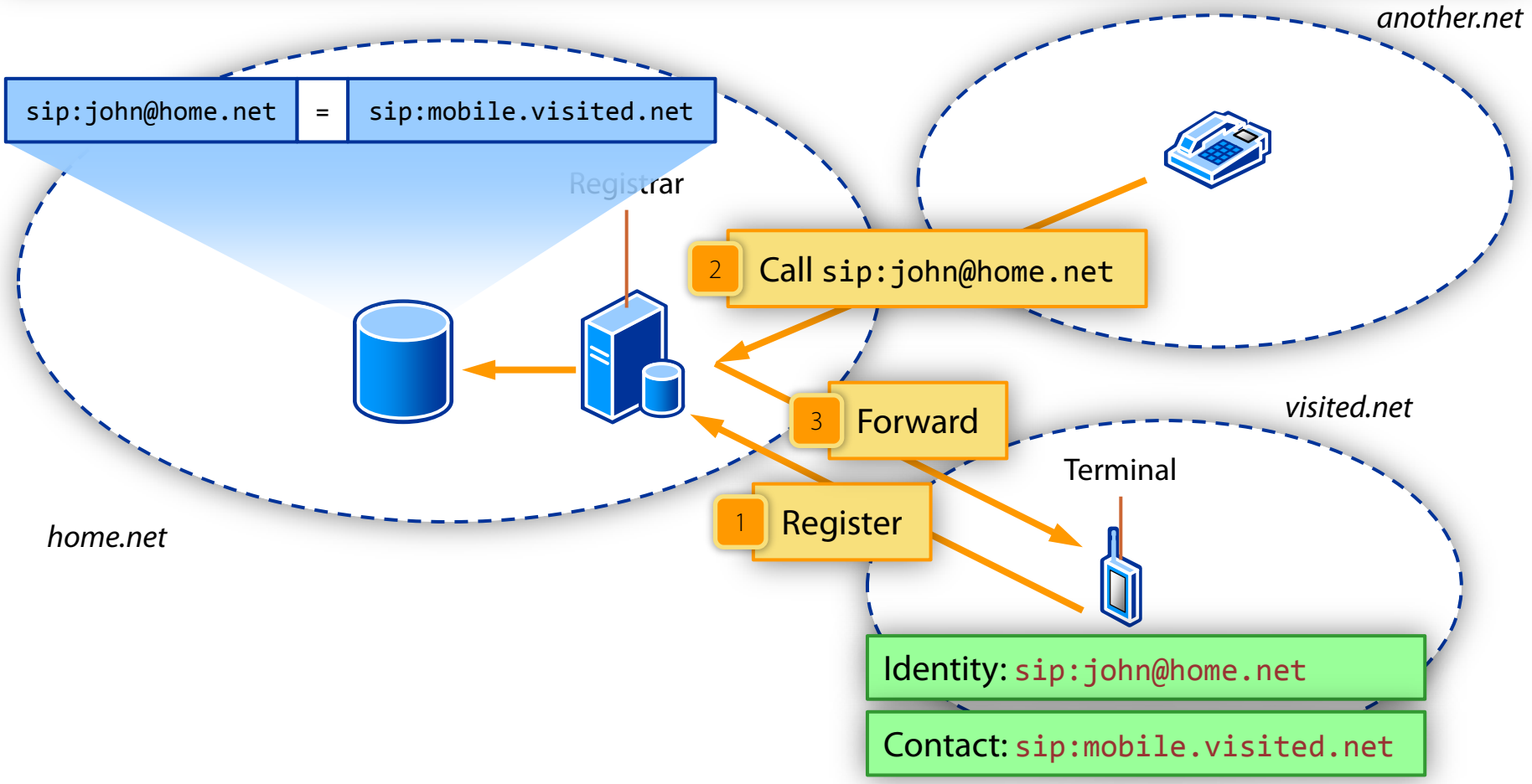
- When registering, a SIP terminal sends its identity and contact information



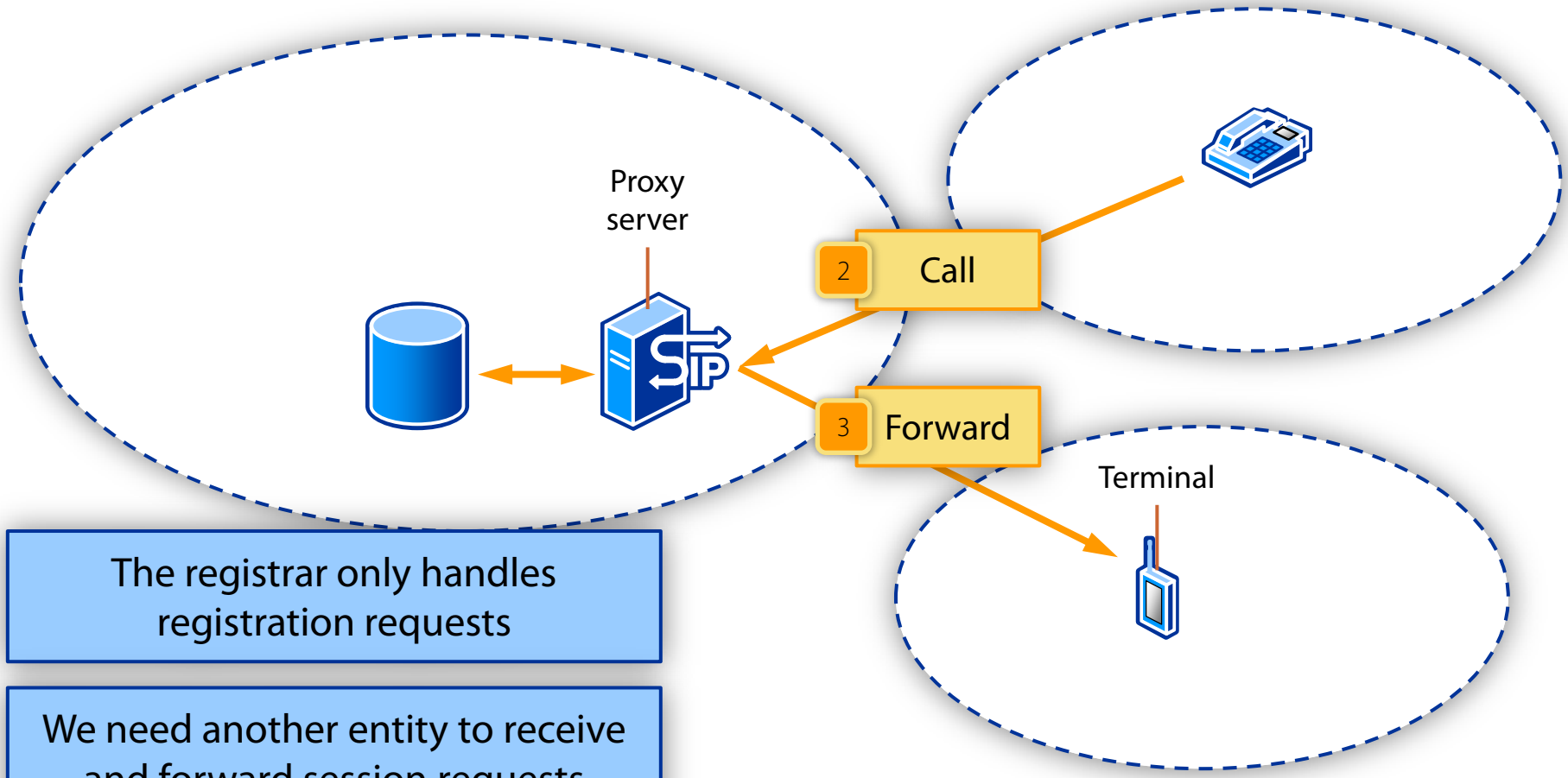
- Terminals are located solely using the SIP URI
- The registrar stores the information in a database
- Other SIP servers use the registration data to forward SIP messages
- There is one registrar for the SIP URI domain name

The Registrar

Is an UAS that accepts only registrations.



The Proxy

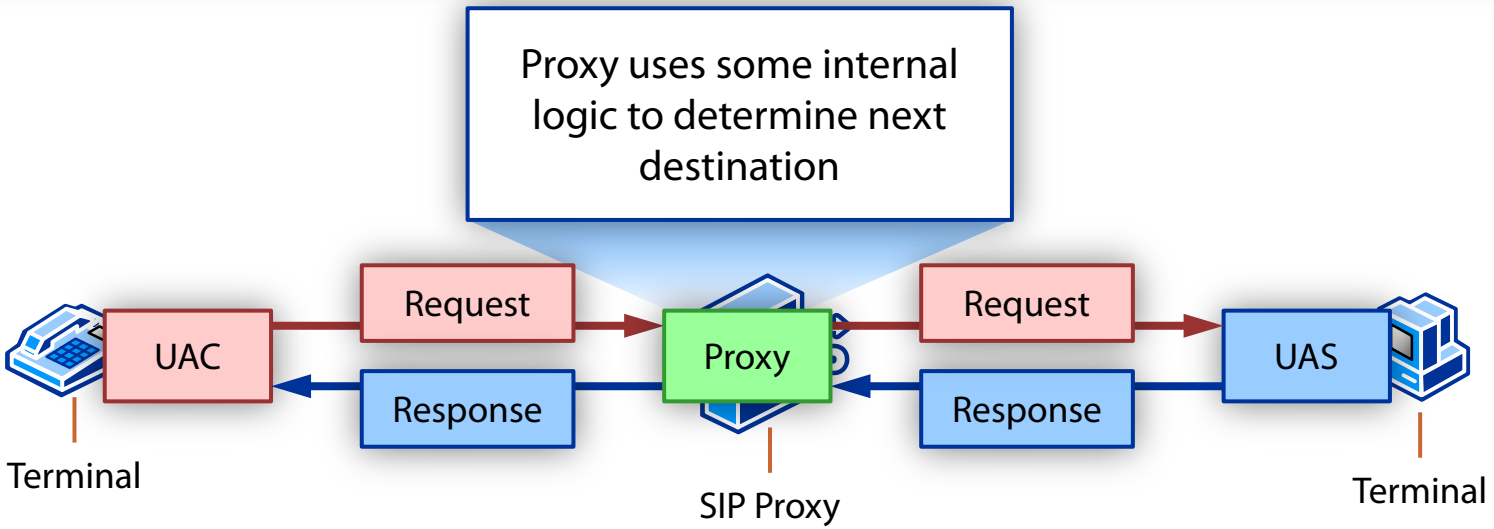


The registrar only handles registration requests

We need another entity to receive and forward session requests

The Proxy

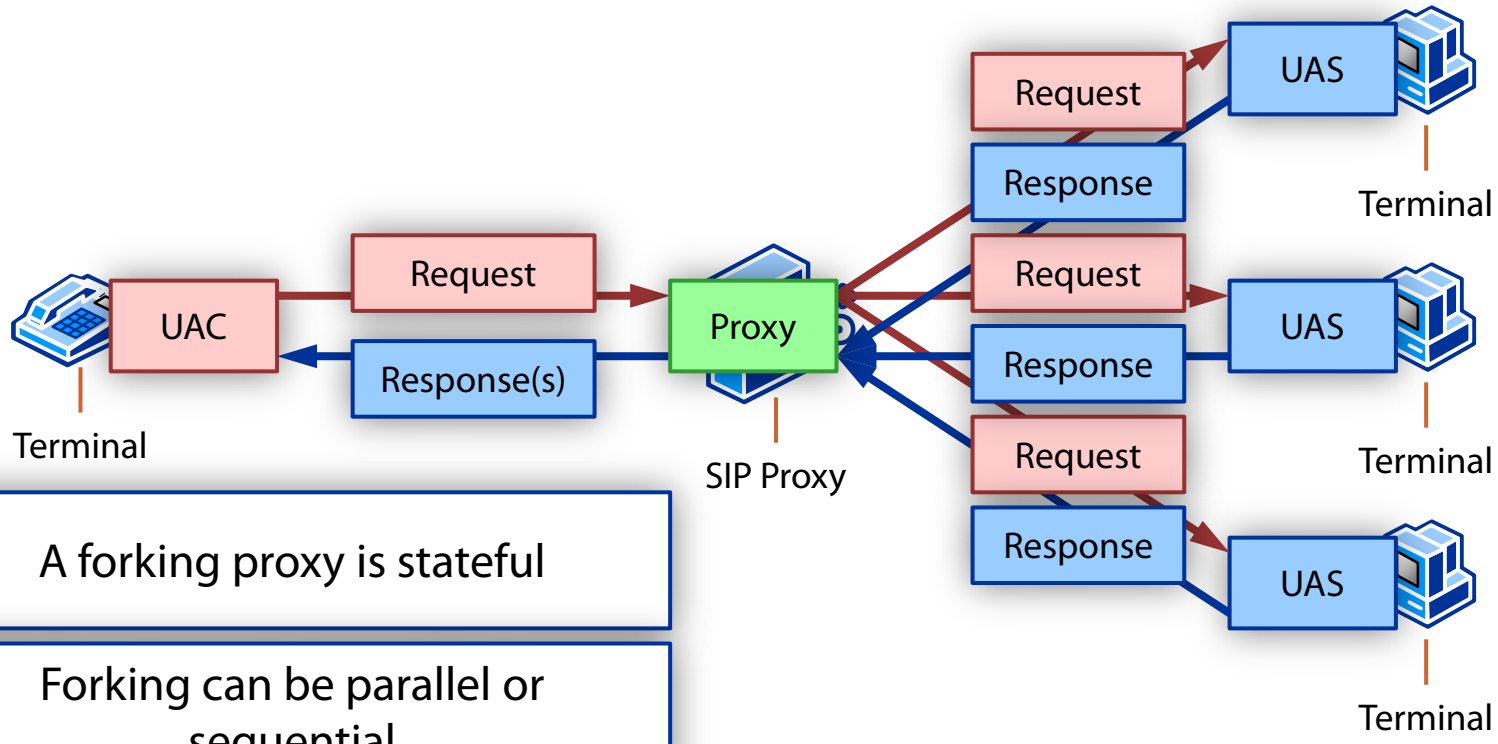
Receives and forwards (proxies) SIP requests and responses



- Proxy types
  - **Stateless** — Does not remember state, simply forwards SIP messages
  - **Transaction stateful** — Remembers state during a transaction: request - response
  - **Call stateful** — Remembers state during an entire session (dialog)

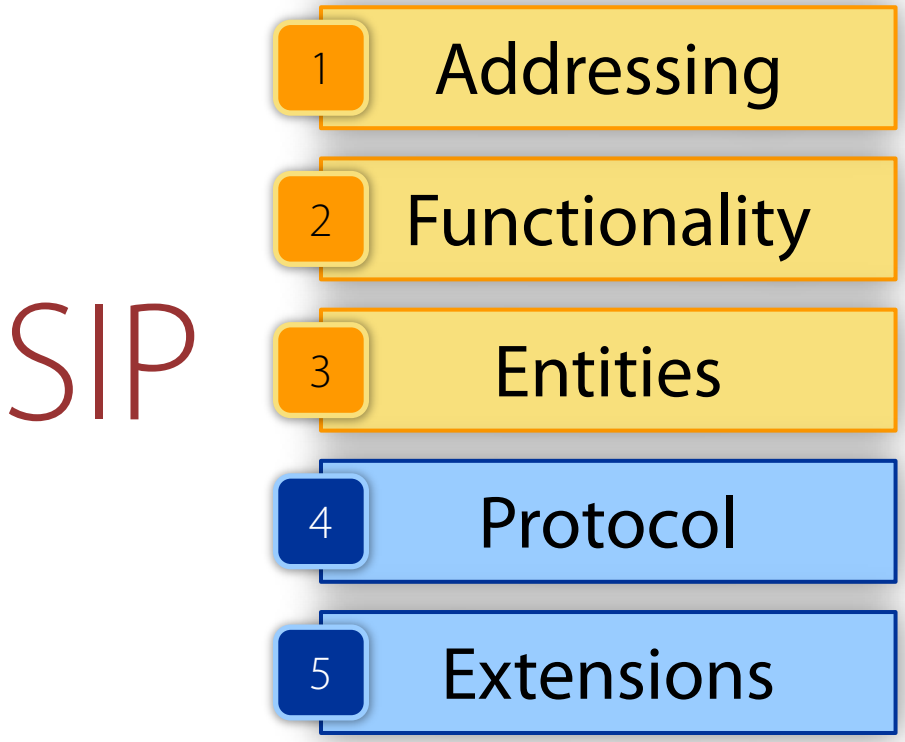
The Forking Proxy

A proxy that supports multiple destinations



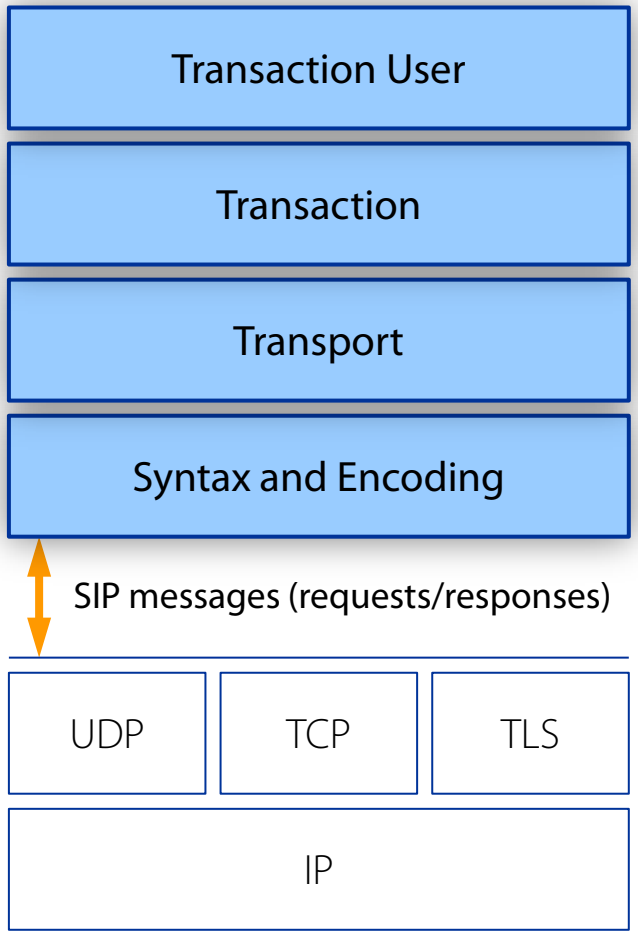
1. A forking proxy is stateful
2. Forking can be parallel or sequential
3. The proxy may return multiple responses

Let's Recap



Functional Layers

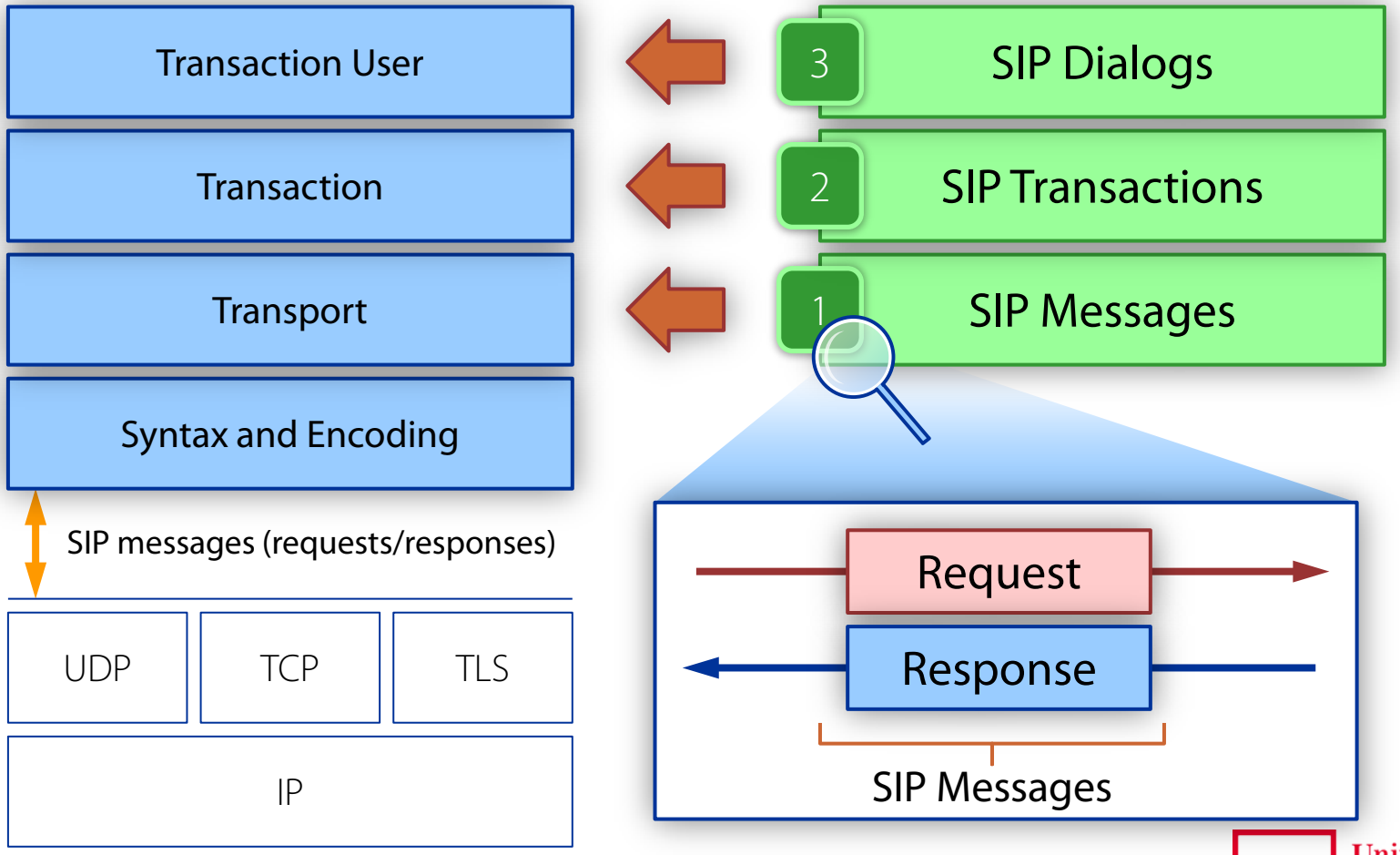
SIP is structured as a layered protocol



- Management of SIP sessions
- Handles retransmissions, duplicates, timeouts
- Matches requests to responses
- Rules for sending and receiving SIP messages
- Different for UAC and UAS
- Message encoding and parsing
- Augmented Backus–Naur Form grammar

Functional Layers

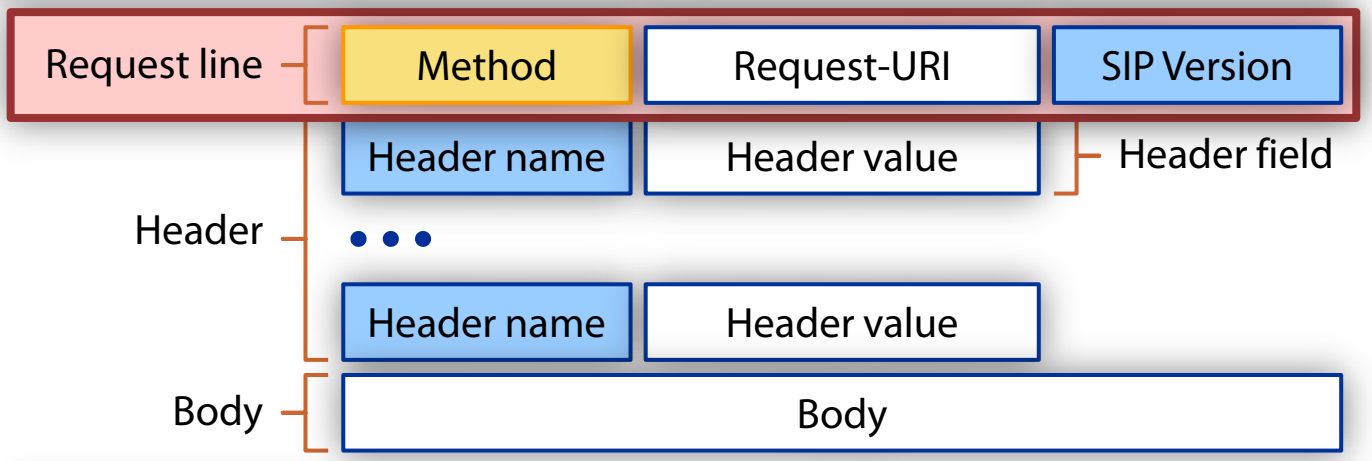
SIP is structured as a layered protocol



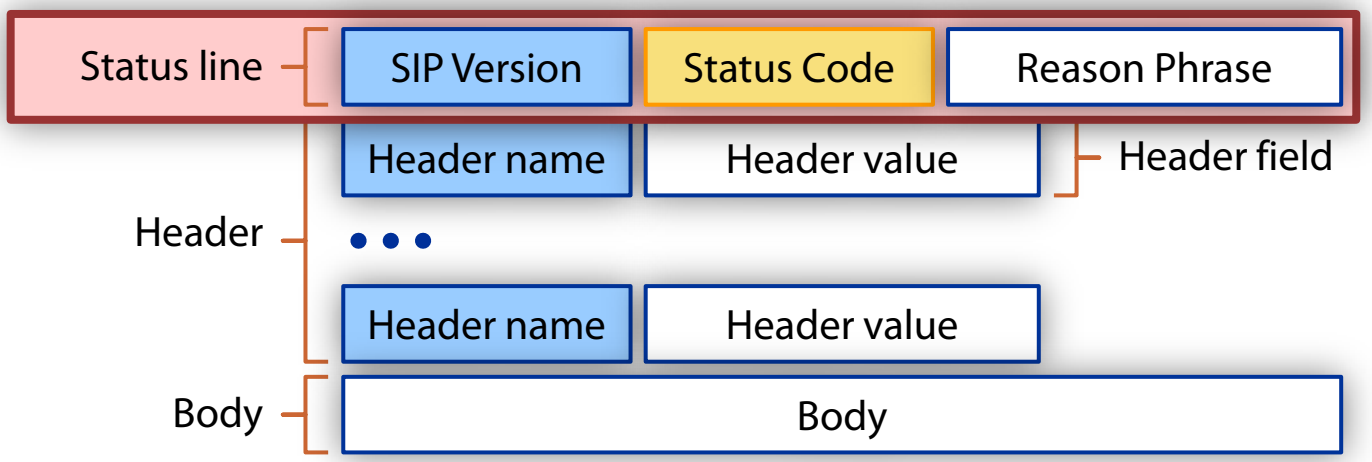
SIP Messages

SIP is a text-based protocol, all SIP messages are in plain-text

1 Request

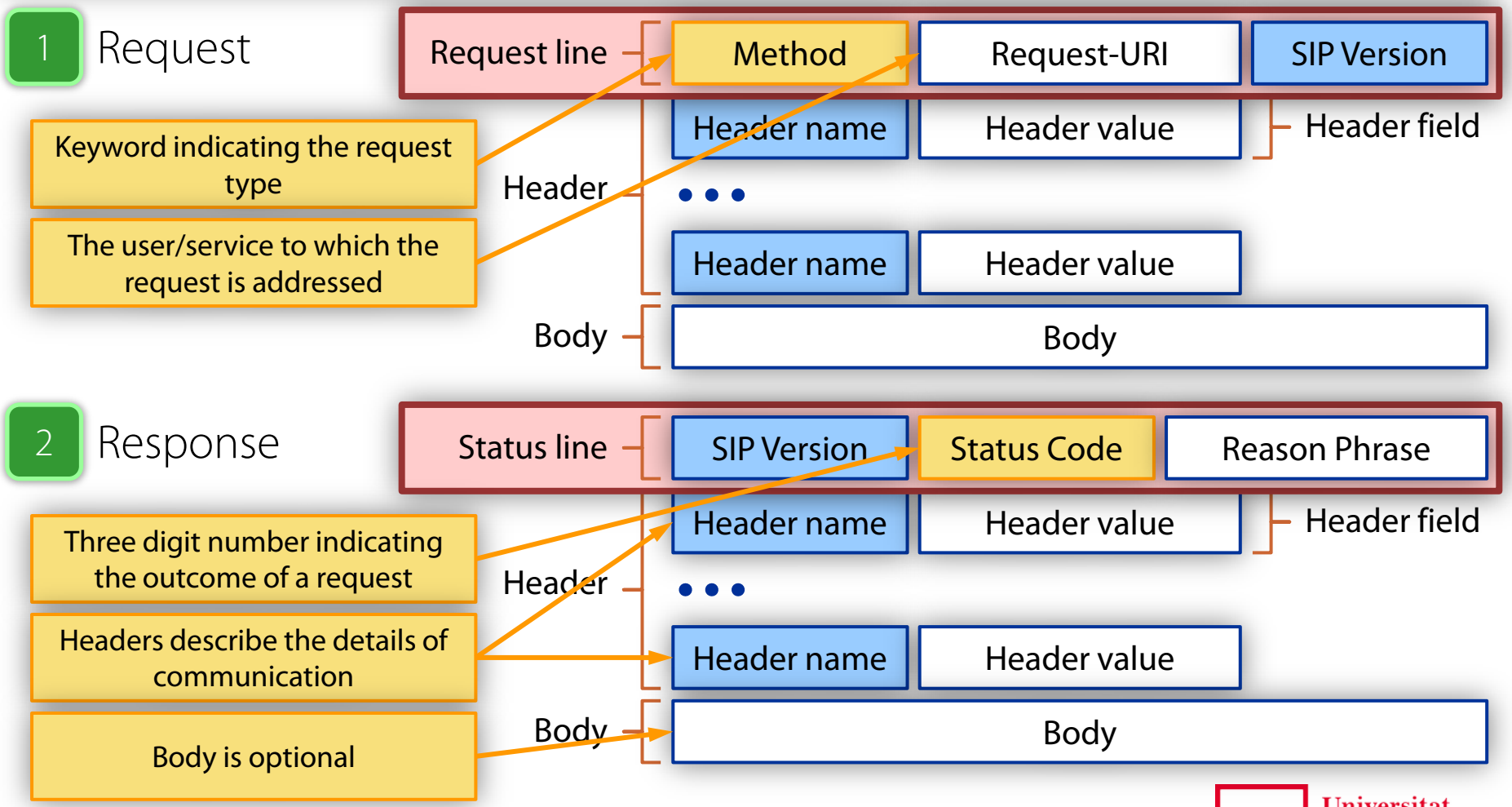


2 Response



SIP Messages

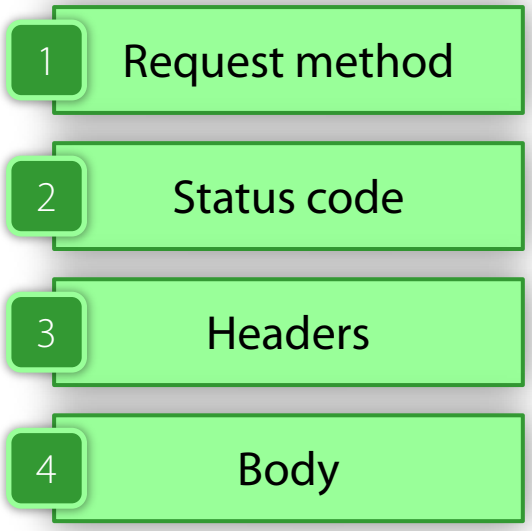
SIP is a text-based protocol, all SIP messages are in plain-text



SIP Messages

- Let's summarize

SIP Message



SIP Methods

The base standard (RFC 3261) defines six SIP methods

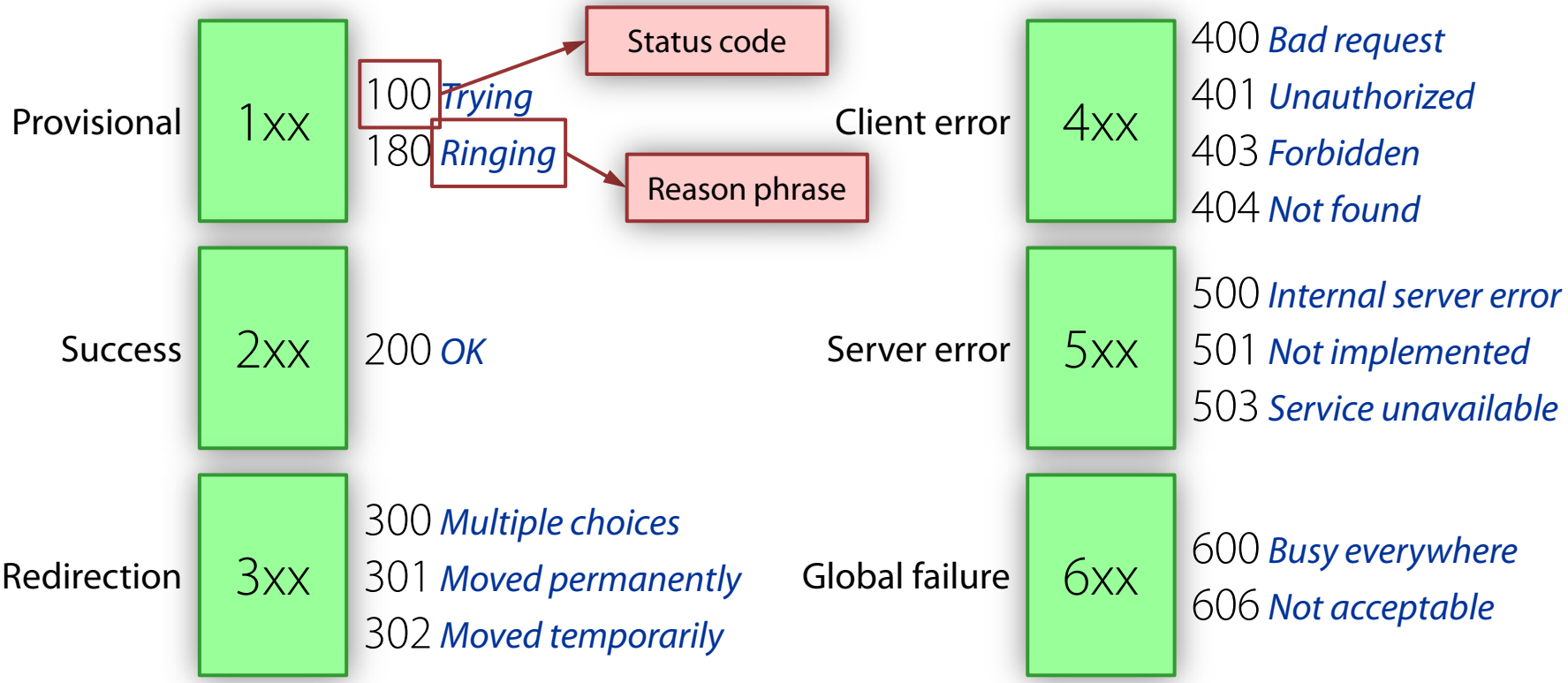
- Extension standards define additional methods
- For now let's focus on the first six

1. **REGISTER**
   - Used during registration and sent to the registrar
   - Maps a SIP identity to the contact address (IP address)
2. **INVITE**
   - Invitation to participate in (establish) a session
   - The body must include a description of the session
3. **ACK**
   - Final acknowledgment for a session establishment
   - Concludes an INVITE transaction
4. **BYE**
   - Terminates an established session
5. **CANCEL**
   - Cancels an in-progress request
   - Does not affect requests that received final responses
6. **OPTIONS**
   - Queries servers for their capabilities

SIP Status Codes

Used in responses to indicate the outcome of a request

- Three digit number, may be followed by a human readable phrase



SIP Status Codes

Used in responses to indicate the outcome of a request

- Let's summarize the status codes

Provisional	1xx	• Request received, continuing to process the request
Success	2xx	• Request was successfully received, understood, and accepted
Redirection	3xx	• Further action needs to be taken in order to complete the request
Client error	4xx	• The request contains bad syntax or cannot be fulfilled at this server
Server error	5xx	• The server failed to fulfill an apparently valid request
Global failure	6xx	• The request cannot be fulfilled at any server

SIP Headers

The SIP headers describe the details of communication

`<HeaderName> : <HeaderValue>`

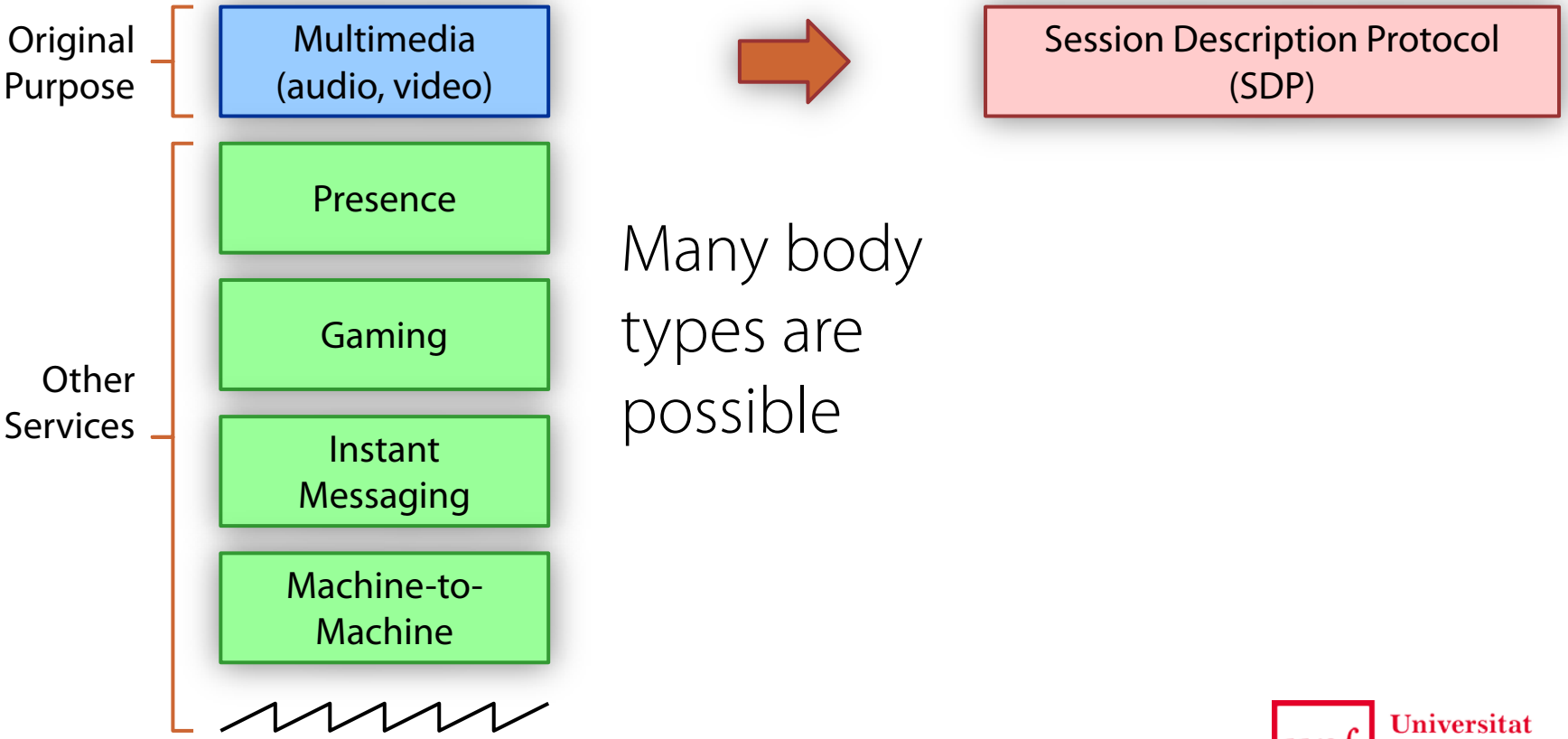
- Let's look at some common headers names (not a complete list)

To	• Identifies the recipient of the request
From	• Identifies the sender of the request
Subject	• Describes the nature of the call
Via	• Indicates the path taken by the request
Call-ID	• Uniquely identifies a SIP session (call)
Content-Length	• The size of the message body in bytes
Content-Type	• The type of the message body
CSeq	• The sequence number of a request message
Route	• The route taken by the SIP message

SIP Body

Carry additional information for the SIP session

- We choose a SIP body suitable for the type of communication (service)
- There exists a special body type for multimedia sessions



Session Description Protocol

SDP A SIP body for multimedia sessions (RFC 4566)
The most common format for describing multimedia sessions

- Text-based, like SIP
- Describes the media streams, connection addresses/ports, codecs, etc.
- All SIP implementations must support SDP

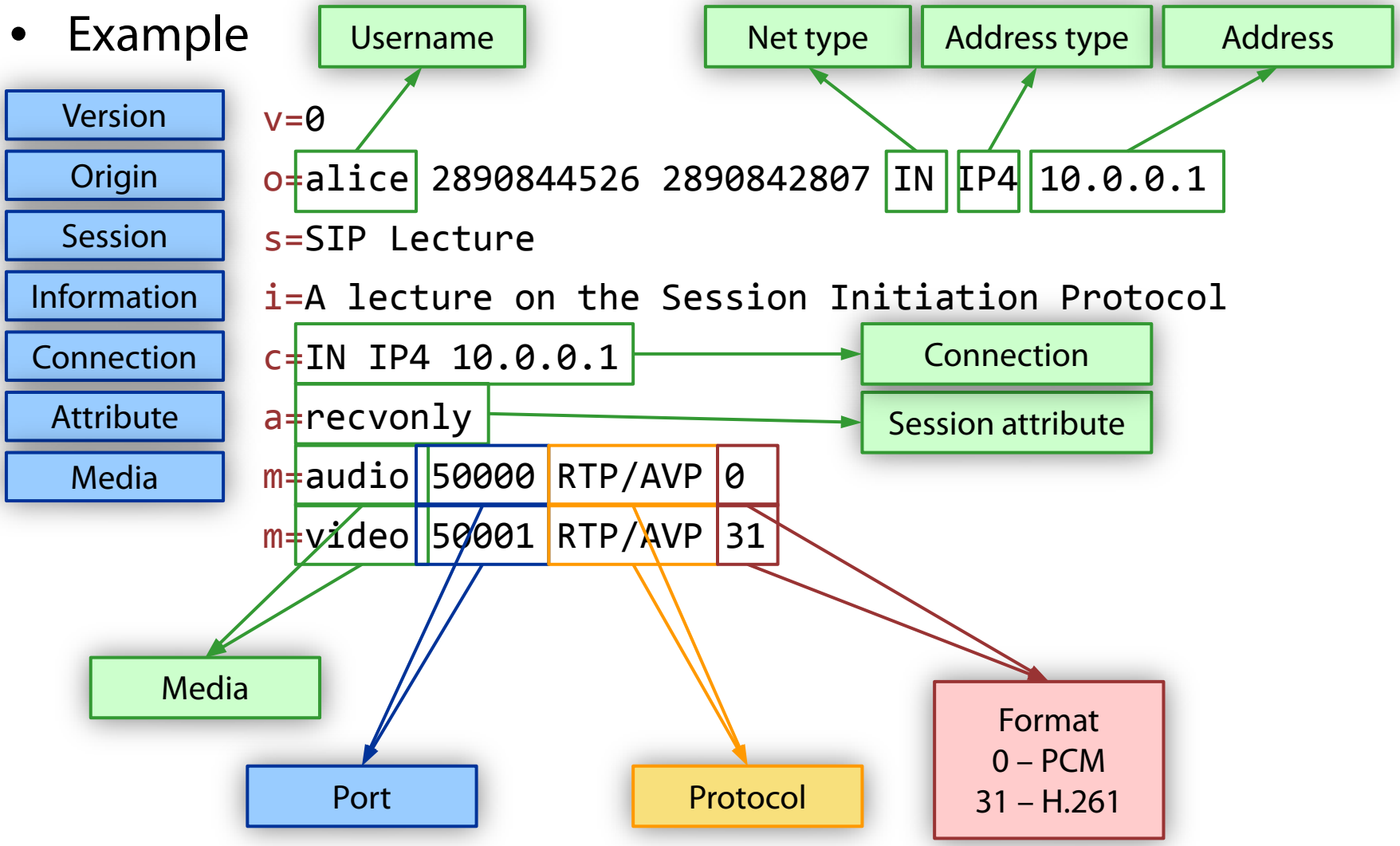
Version	`v=0`
Origin	`o=`<username> <sess-id> <sess-version> <nettype> <addrtype> <unicast-address>
Session	`s=`<session-name>
Information	`i=`<session-information>
Connection	`c=`<nettype> <addrtype> <connection-address>
Attribute	`a=`<attribute>:<value>
Media	`m=`<media> <port> <protocol> <format>

Session level

Media level

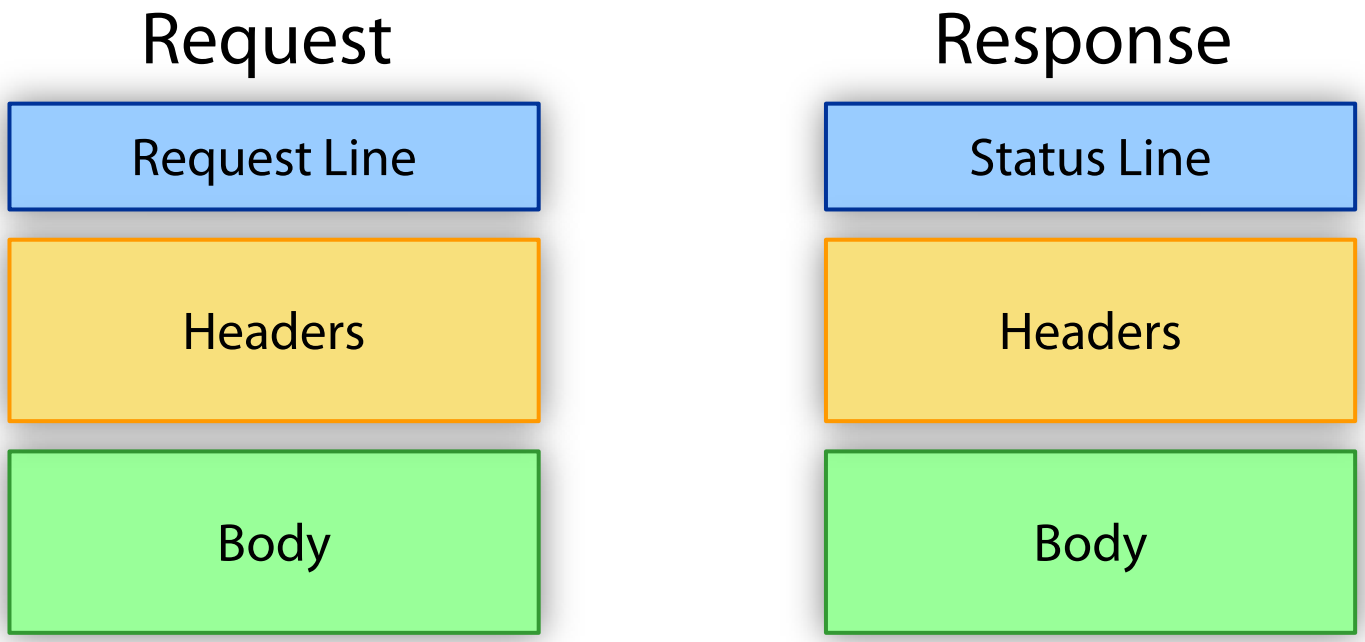
Session Description Protocol

- Example

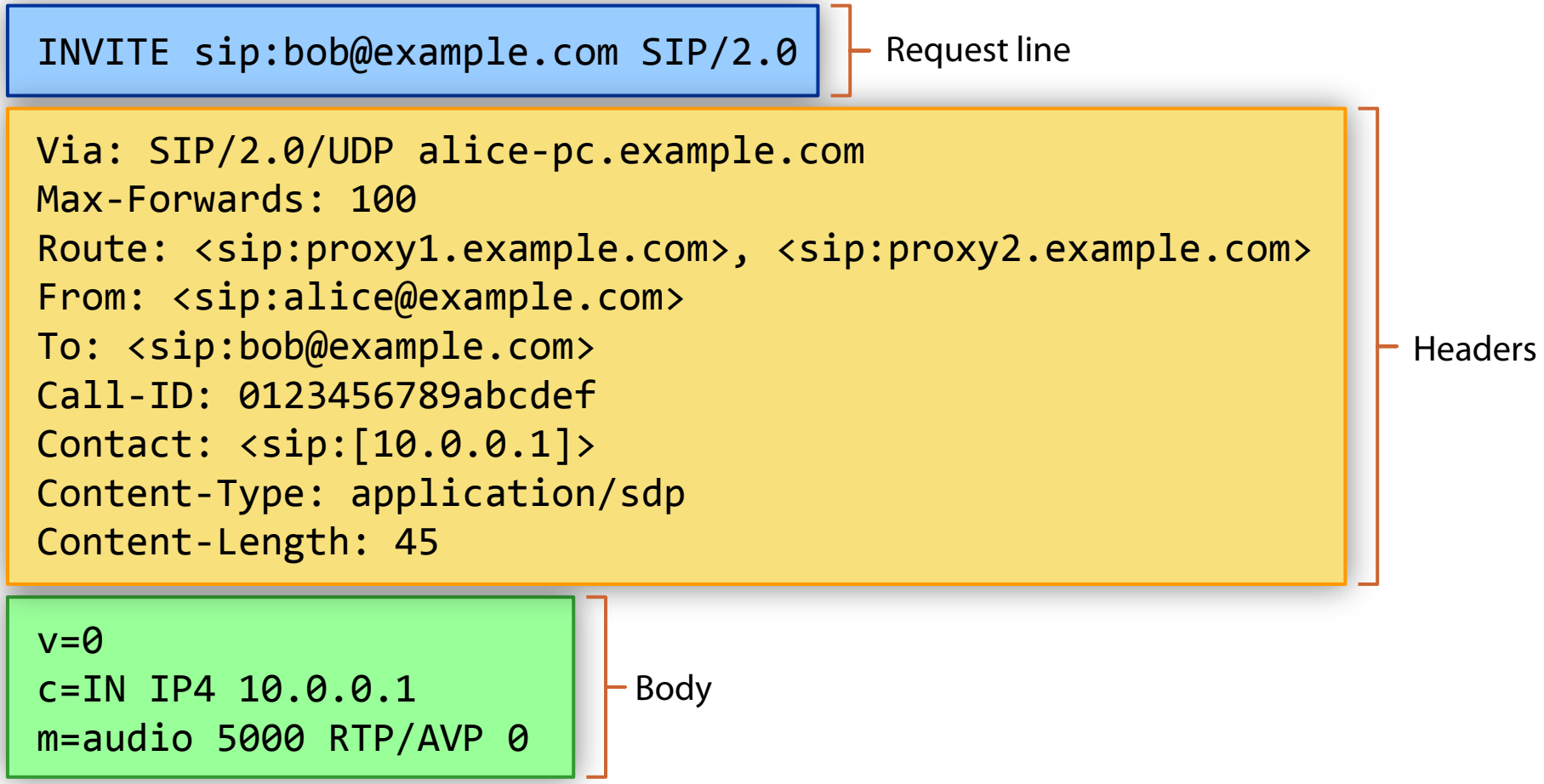


SIP Messages

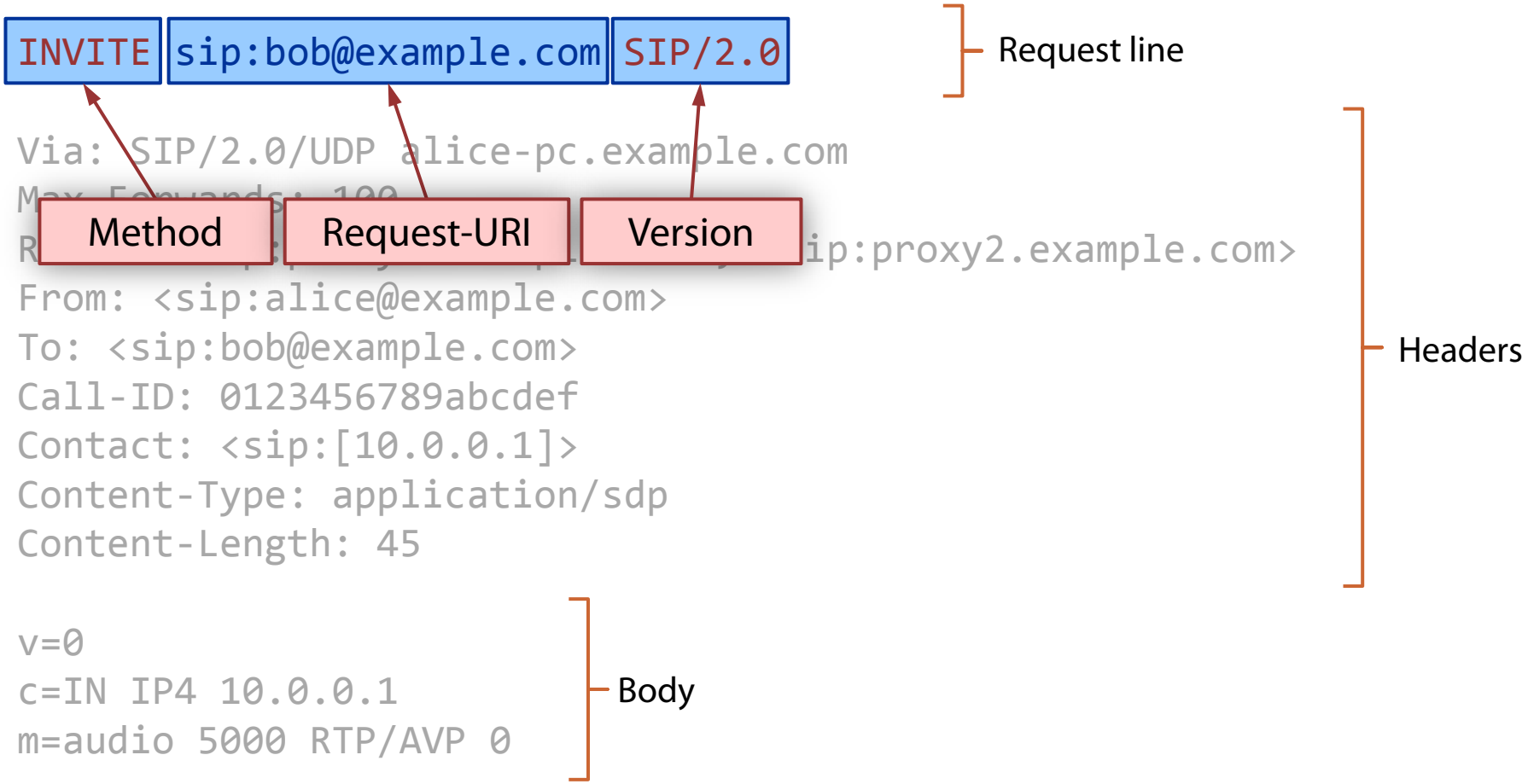
How does a SIP message look like?



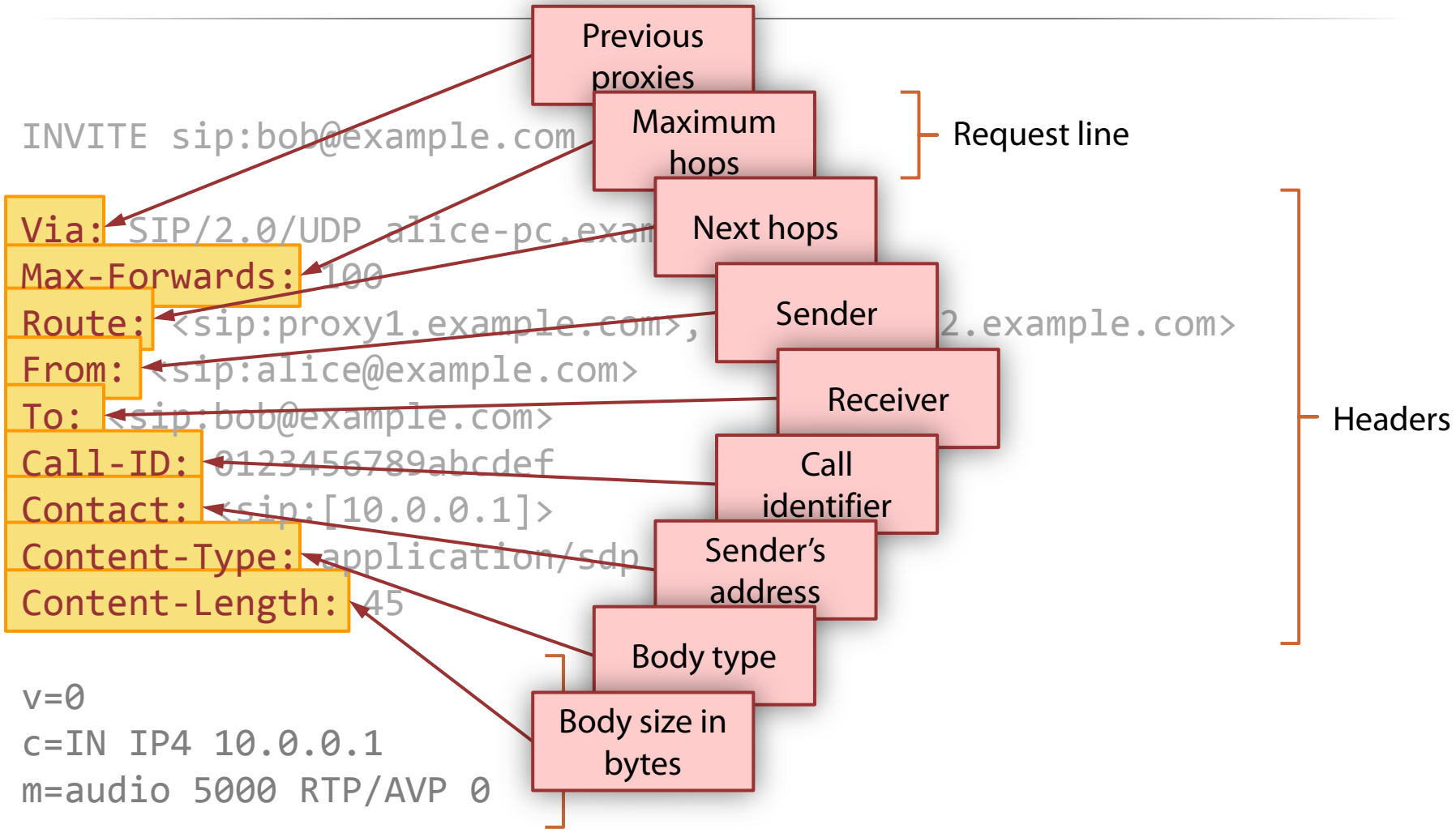
SIP Request Examples



SIP Request Examples



SIP Request Examples

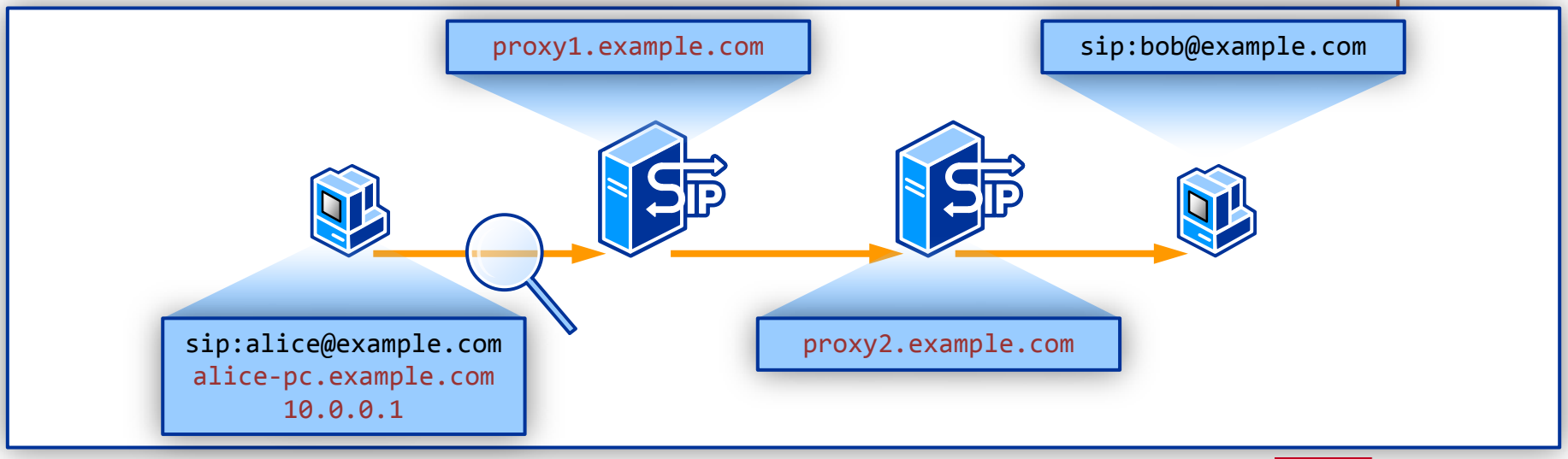


SIP Request Examples

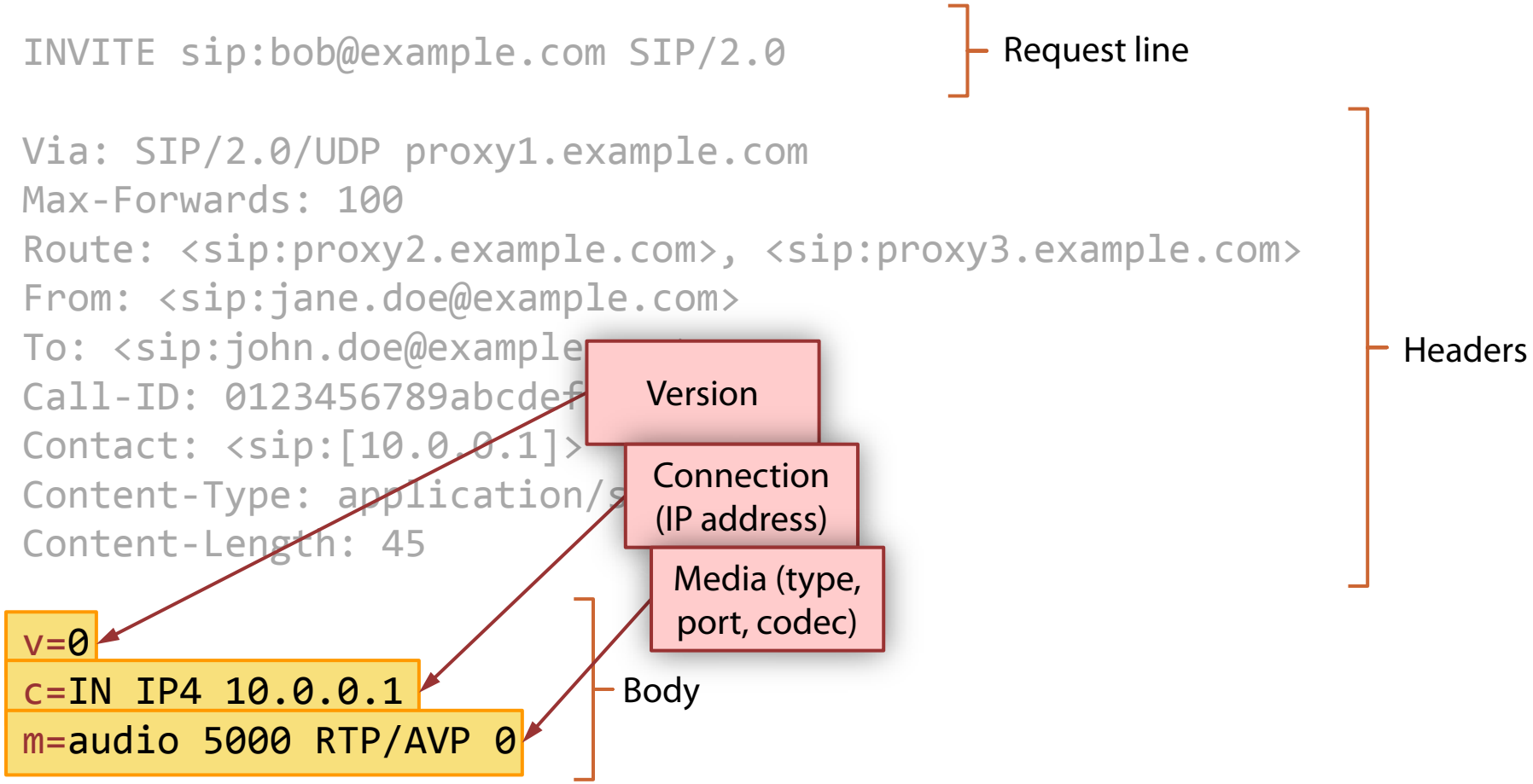
```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/UDP alice-pc.example.com
Max-Forwards: 100
Route: <sip:proxy1.example.com>, <sip:proxy2.example.com>
From: <sip:alice@example.com>
To: <sip:bob@example.com>
Call-ID: 0123456789abcdef
```

Request line

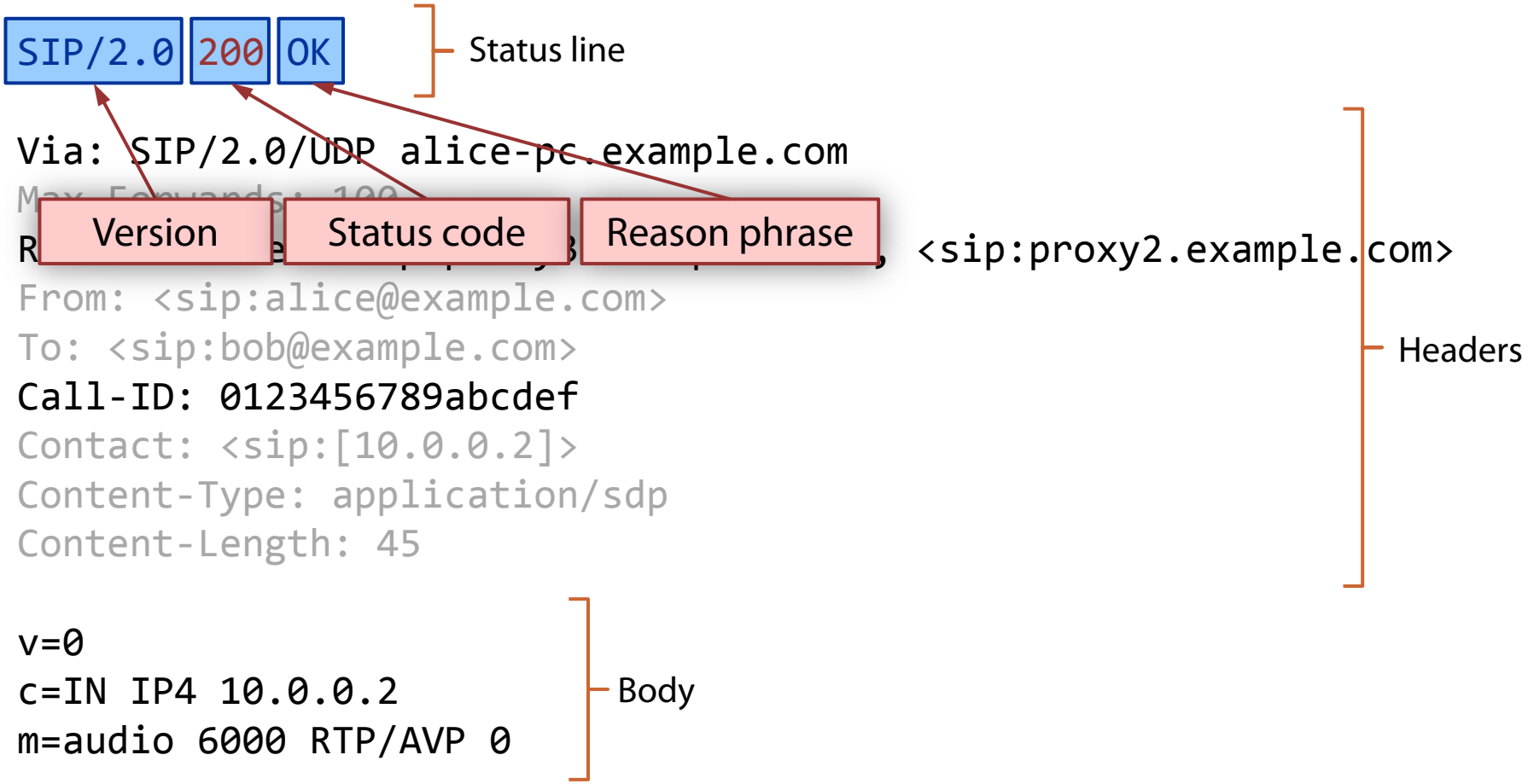
Headers



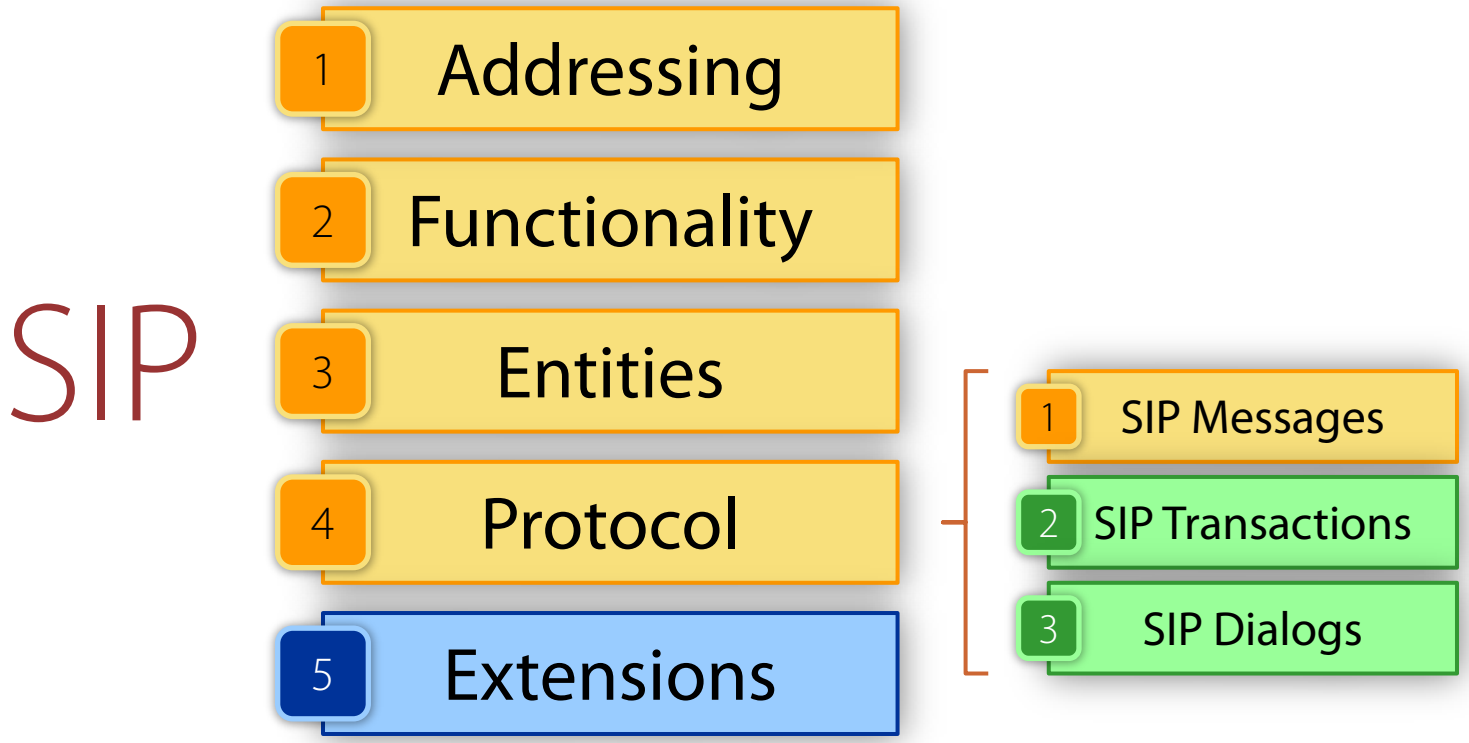
SIP Request Examples



SIP Response Example



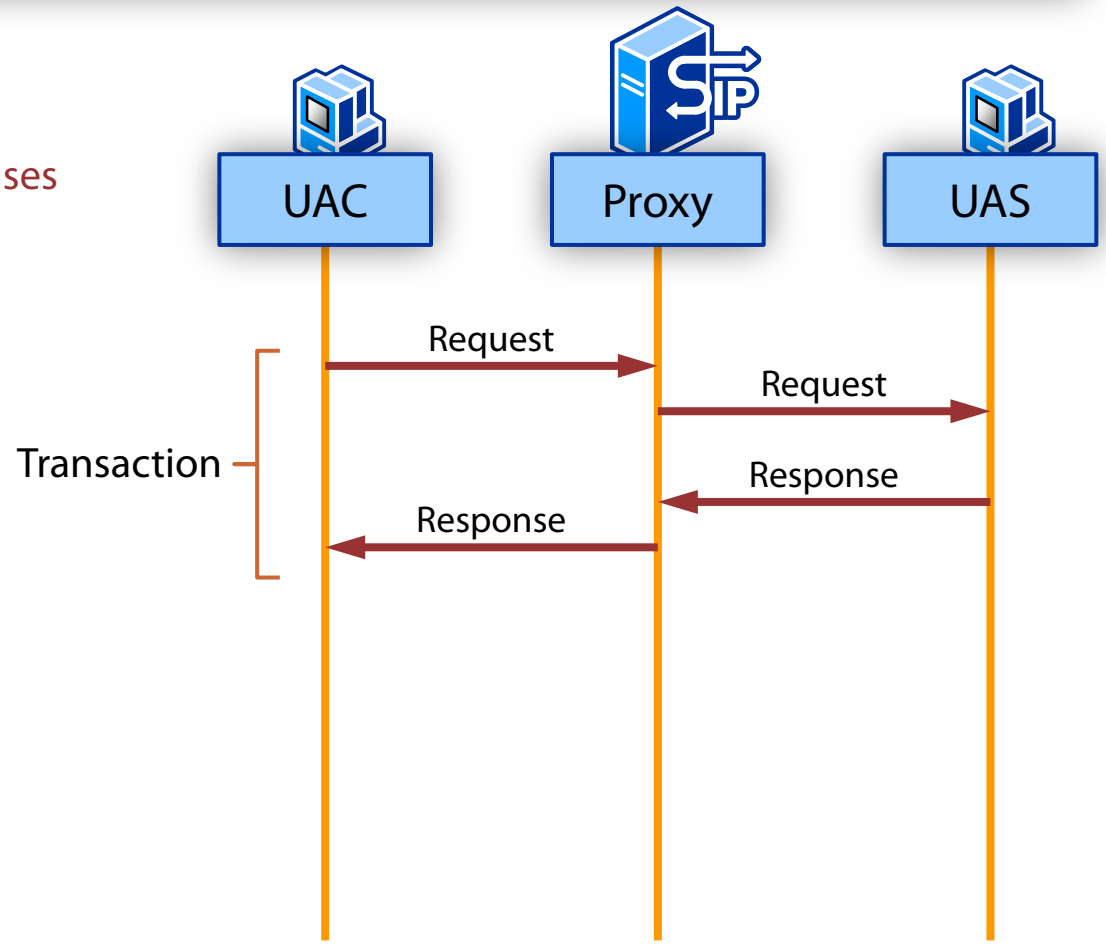
Let's Recap



SIP Transactions

It is a request-response exchange

- Consists of:
  - One request
  - One or more provisional responses
  - One or more final responses



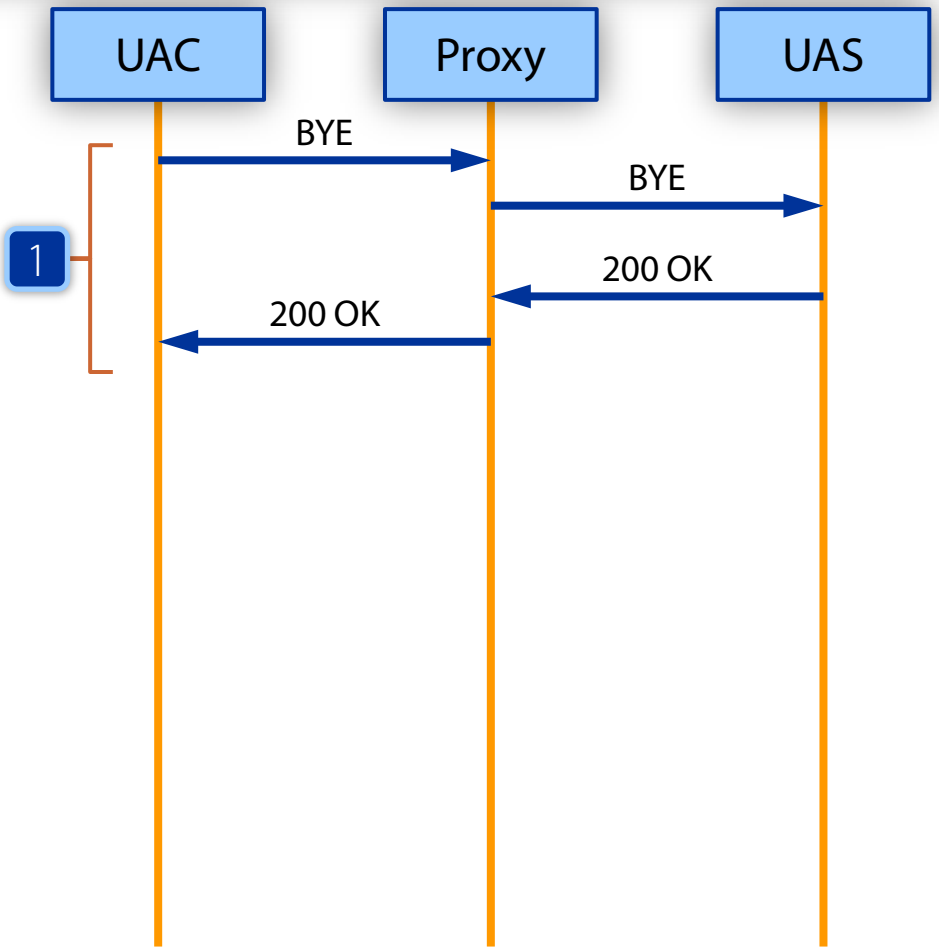
SIP Transactions

It is a request-response exchange

- There are three transaction types

1 Regular transaction

- A transaction with any method other than INVITE, ACK and CANCEL



SIP Transactions

It is a request-response exchange

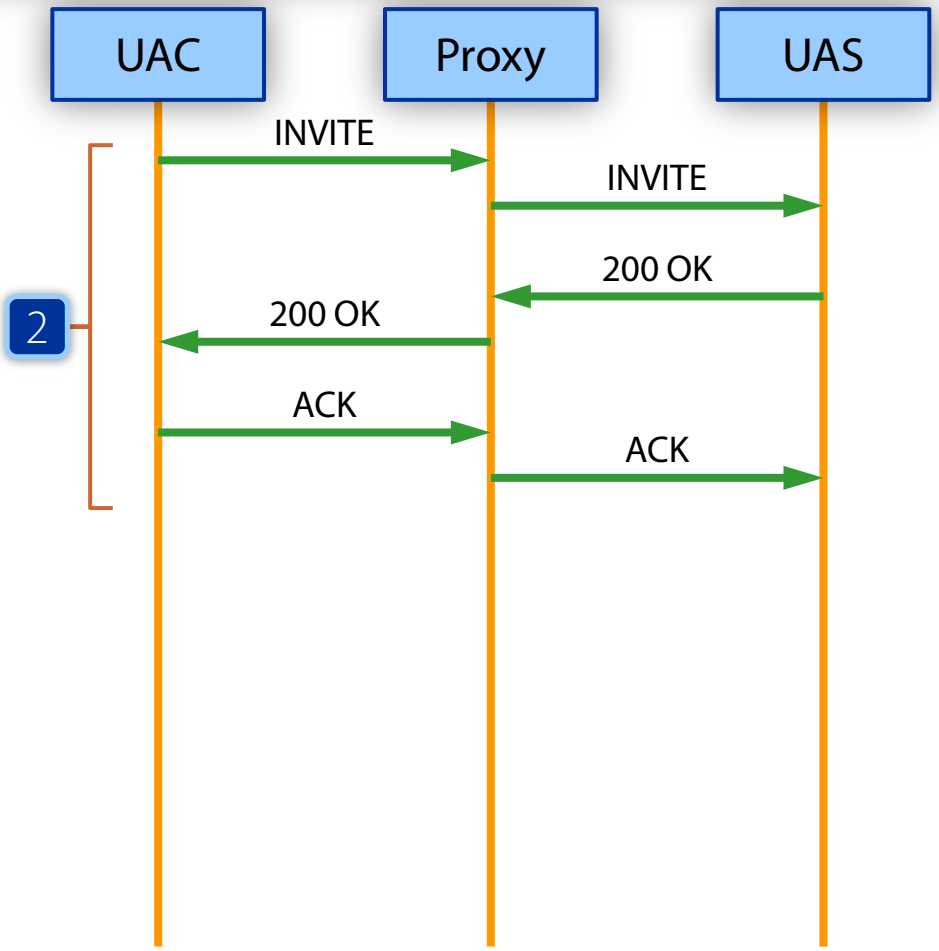
- There are three transaction types

1 Regular transaction

- A transaction with any method other than INVITE, ACK and CANCEL

2 INVITE-ACK transaction

- Involves two transactions INVITE and ACK
- The ACK transaction acknowledges the final response



SIP Transactions

It is a request-response exchange

- There are three transaction types

1 Regular transaction

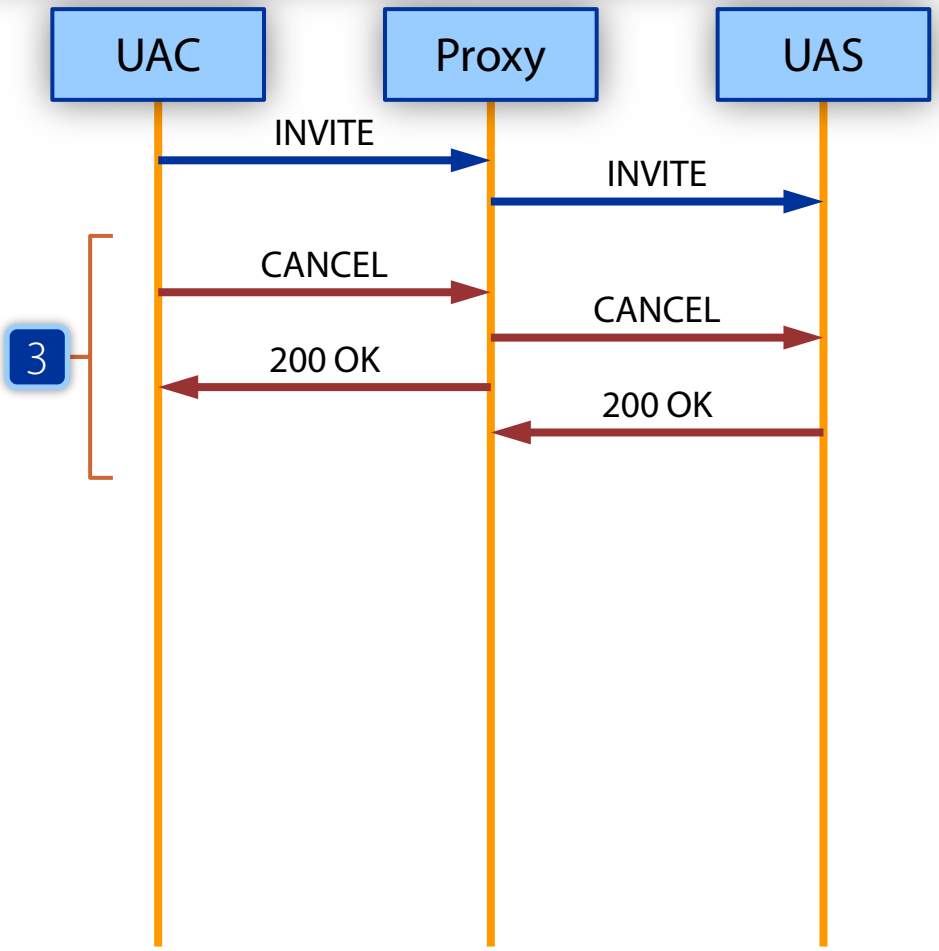
- A transaction with any method other than INVITE, ACK and CANCEL

2 INVITE-ACK transaction

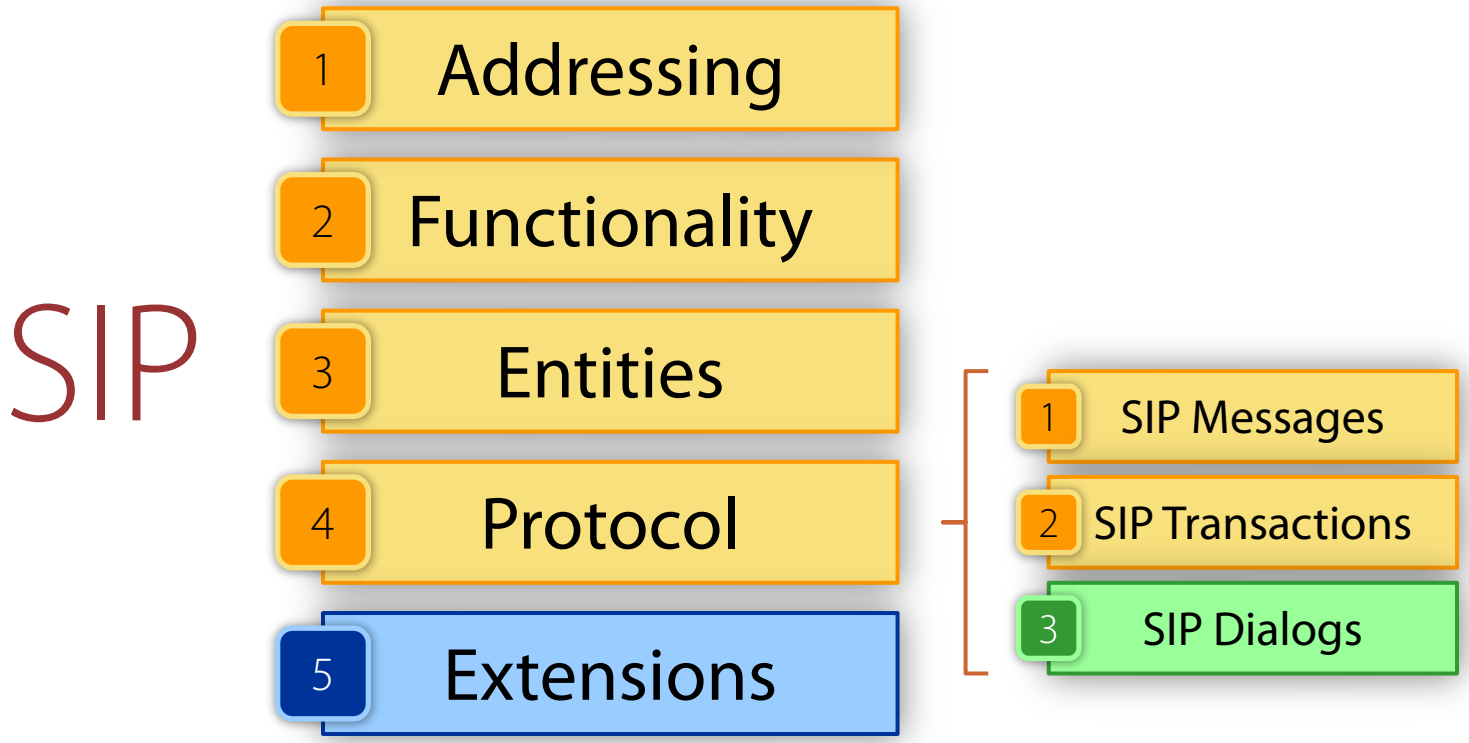
- Involves two transactions INVITE and ACK
- The ACK transaction acknowledges the final response

3 CANCEL transaction

- Connected to a previous transaction
- The reply is generated by the next hop



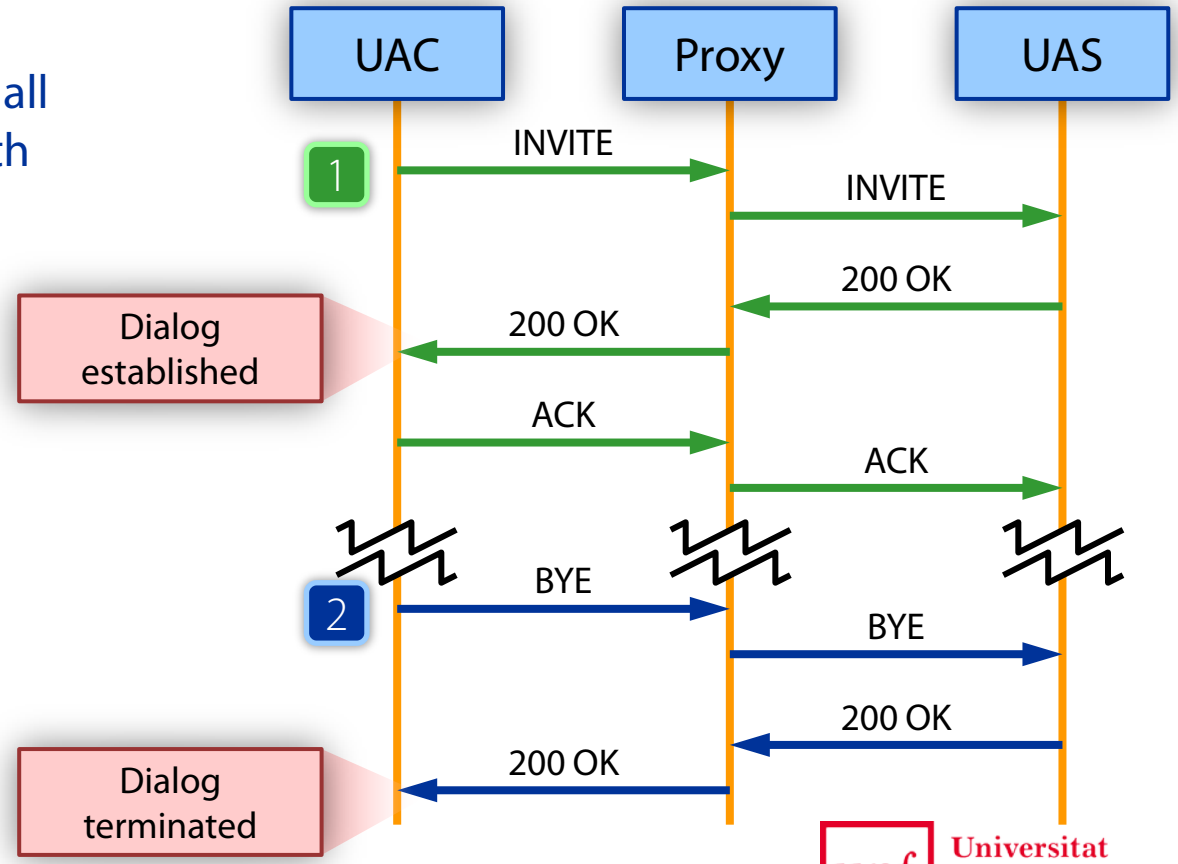
Let's Recap



SIP Dialog

A SIP relationship between two endpoints that persists for some time

- Consists of several SIP transactions
- A SIP multimedia session is an example of a SIP dialog
- Once a dialog is established, all requests follow the same path



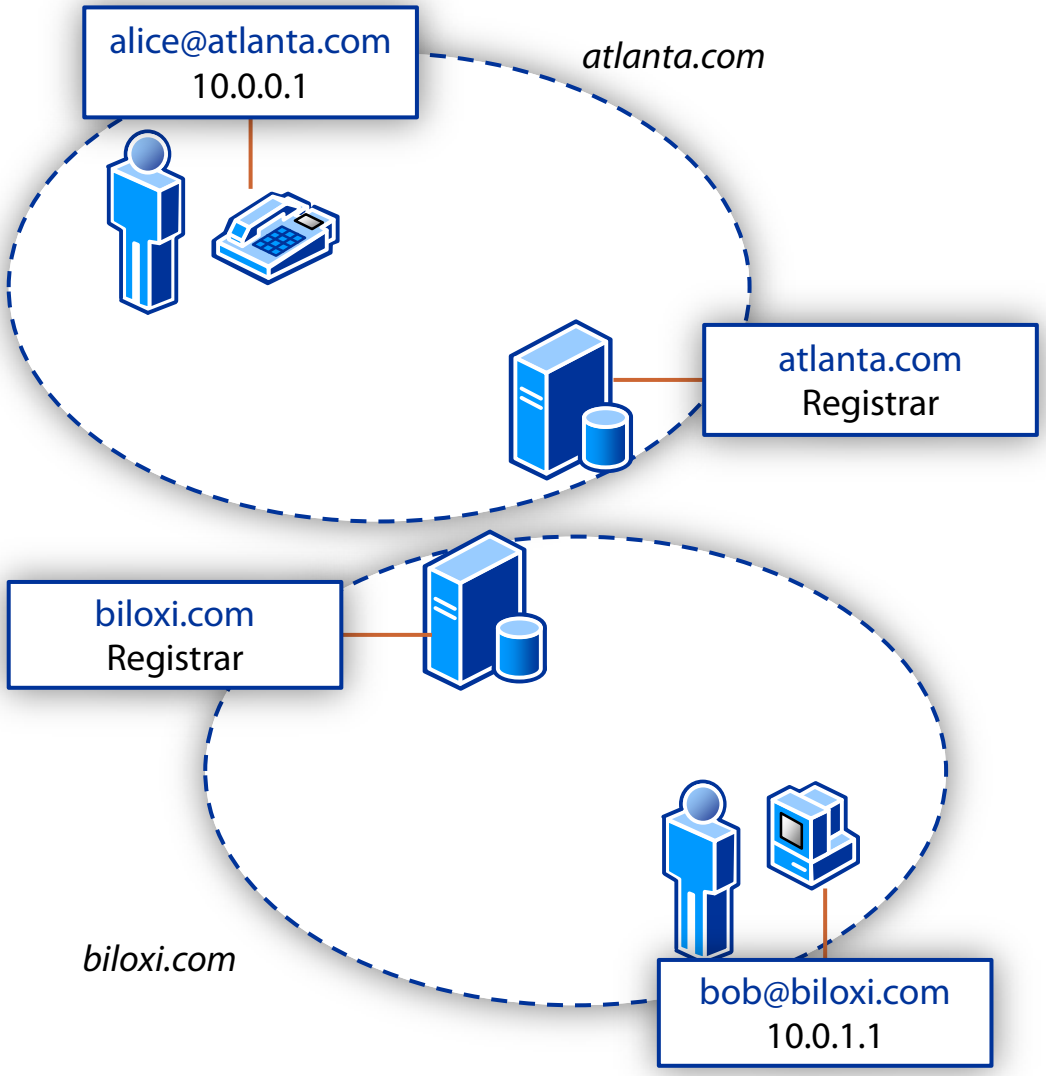
Everything Together

Let's take a detailed look on how SIP works

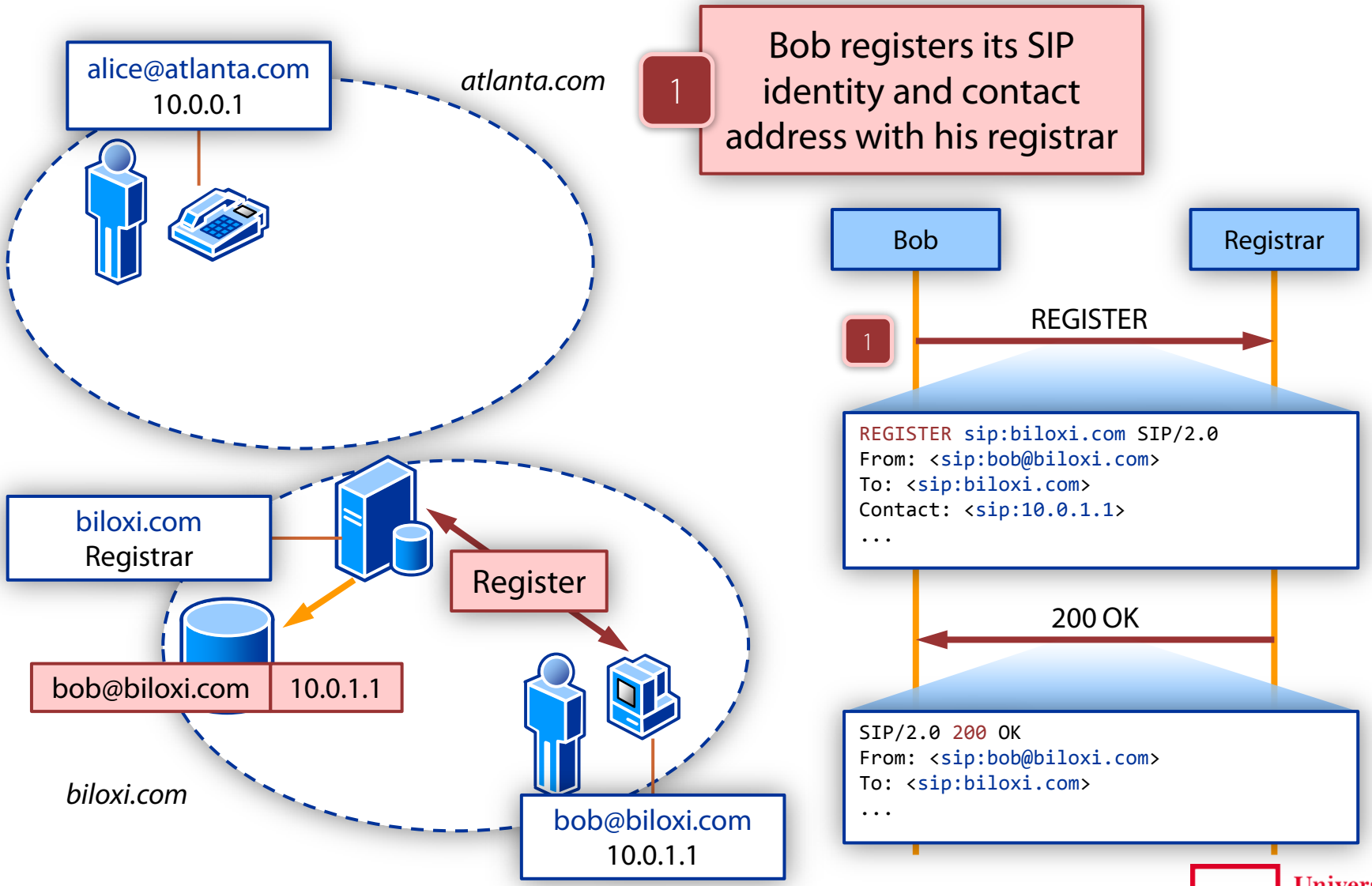
Alice calls Bob

1. Alice is an old user and registered
2. Bob is a new user and first must register

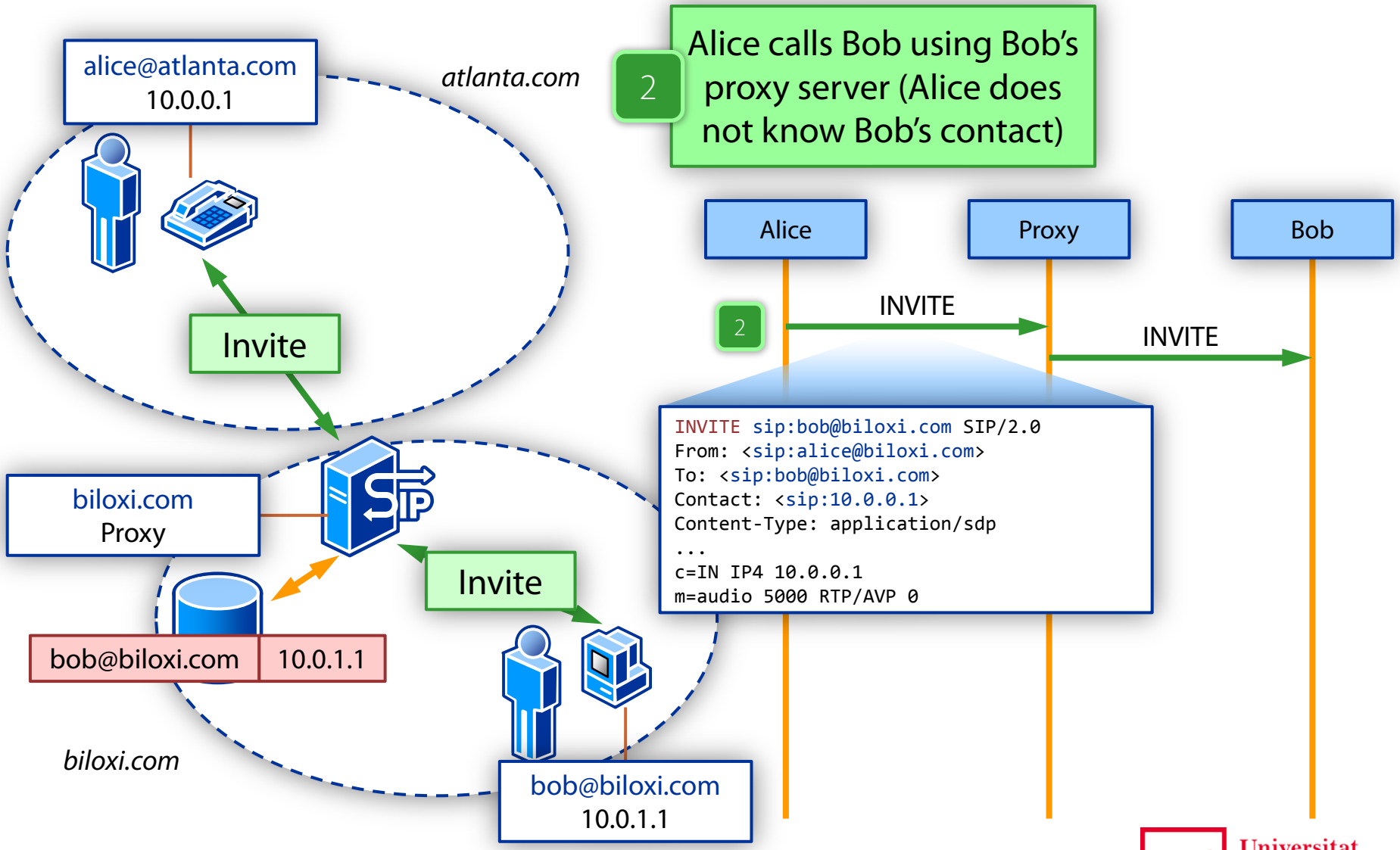
Everything Together



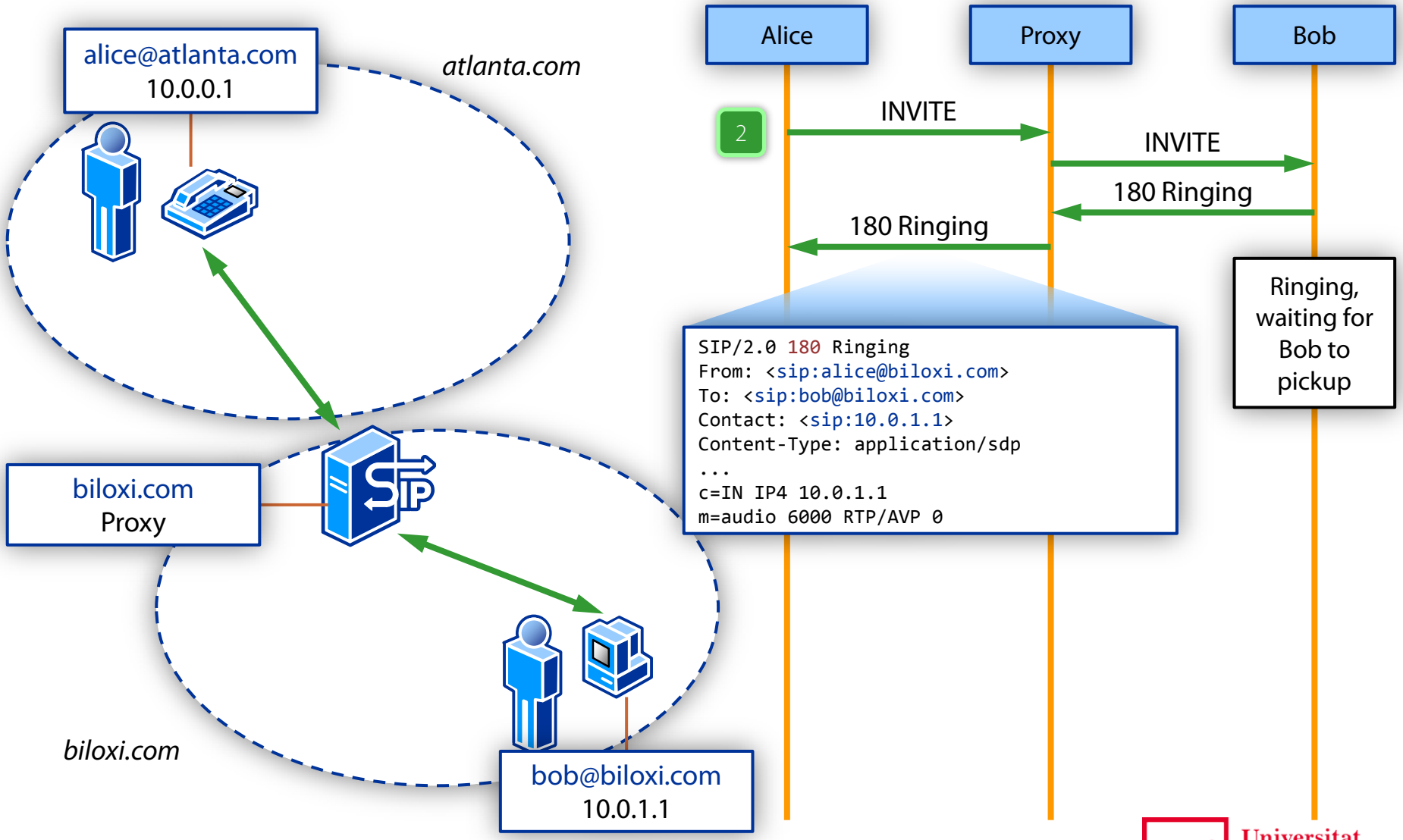
Registration



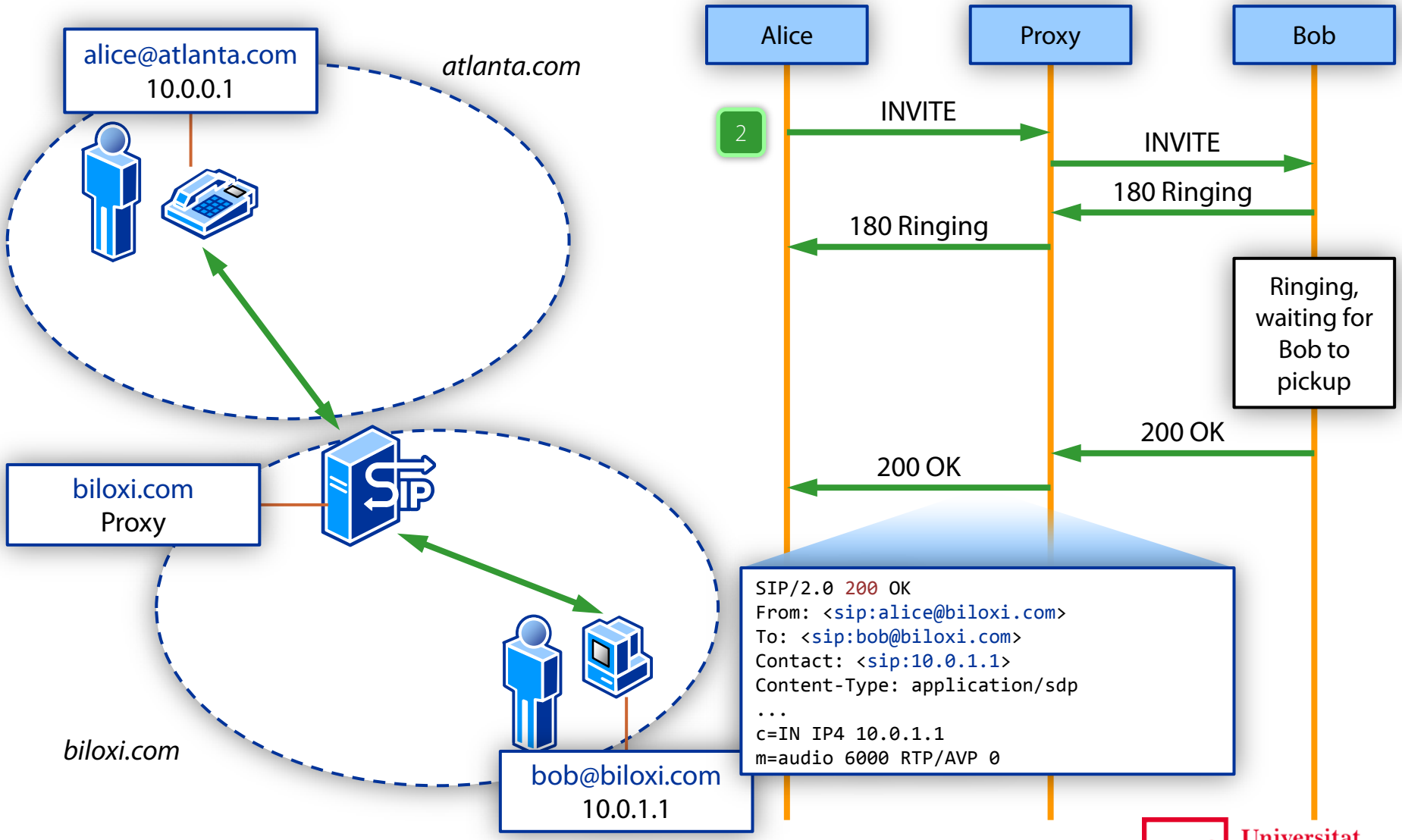
Session Establishment



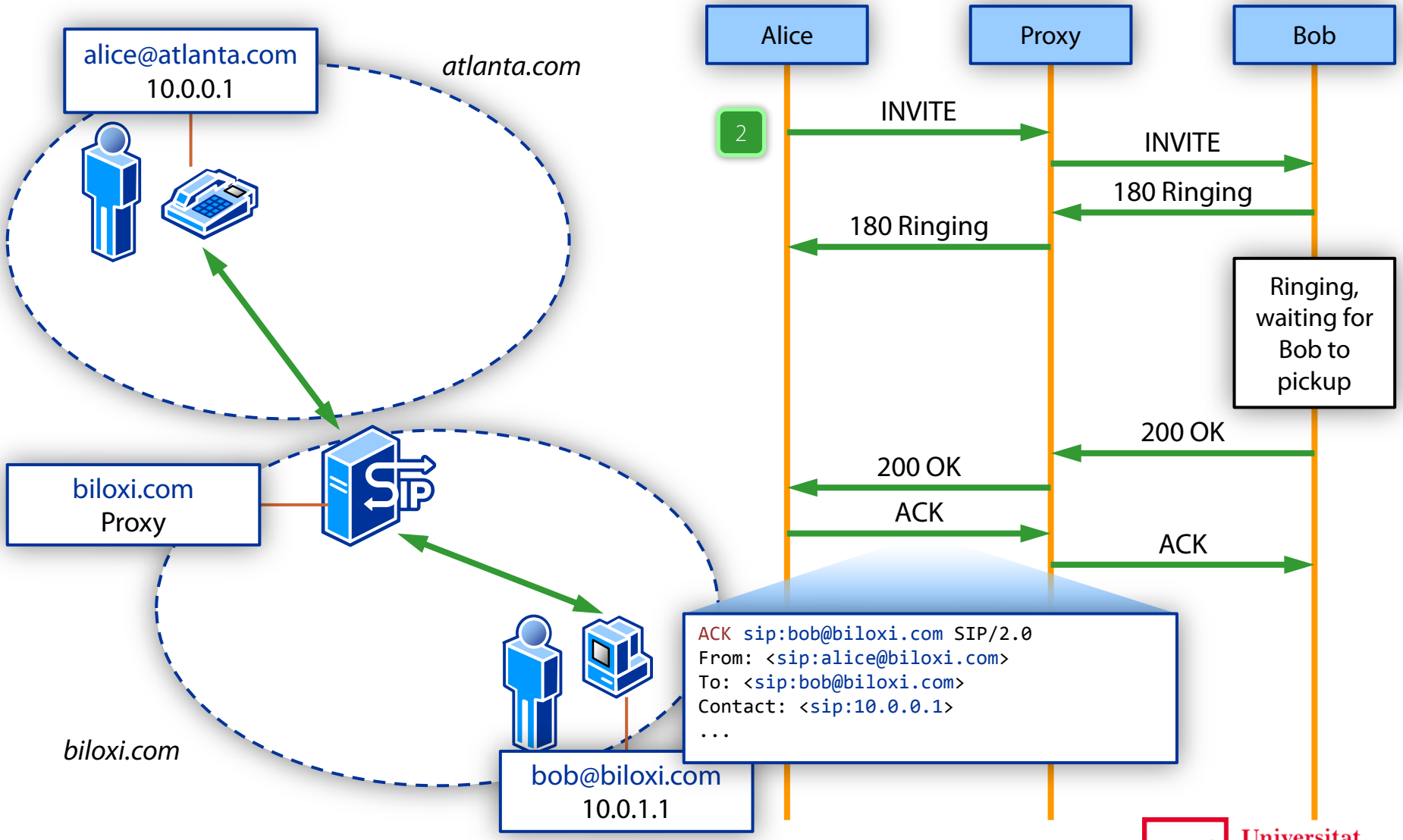
Session Establishment



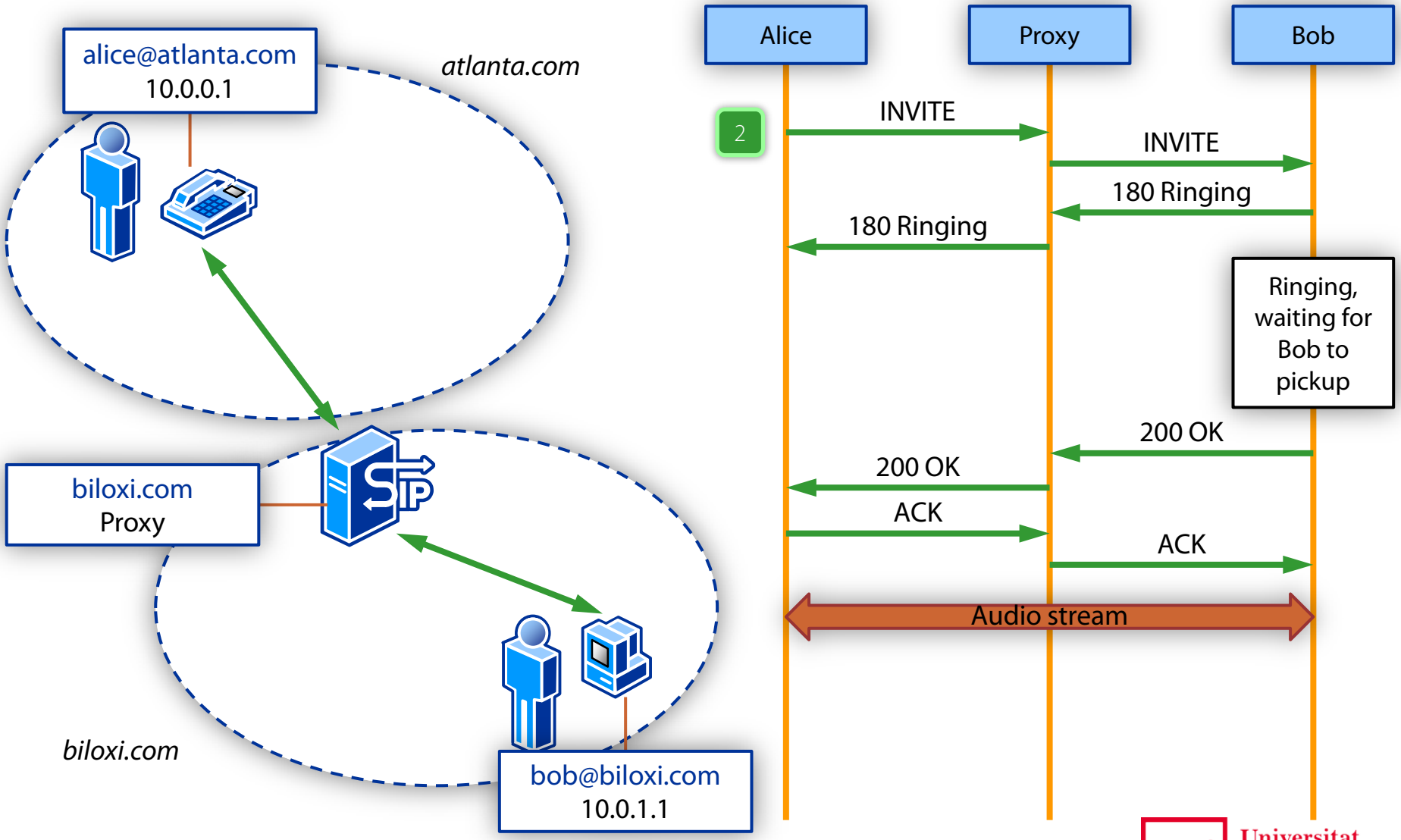
Session Establishment



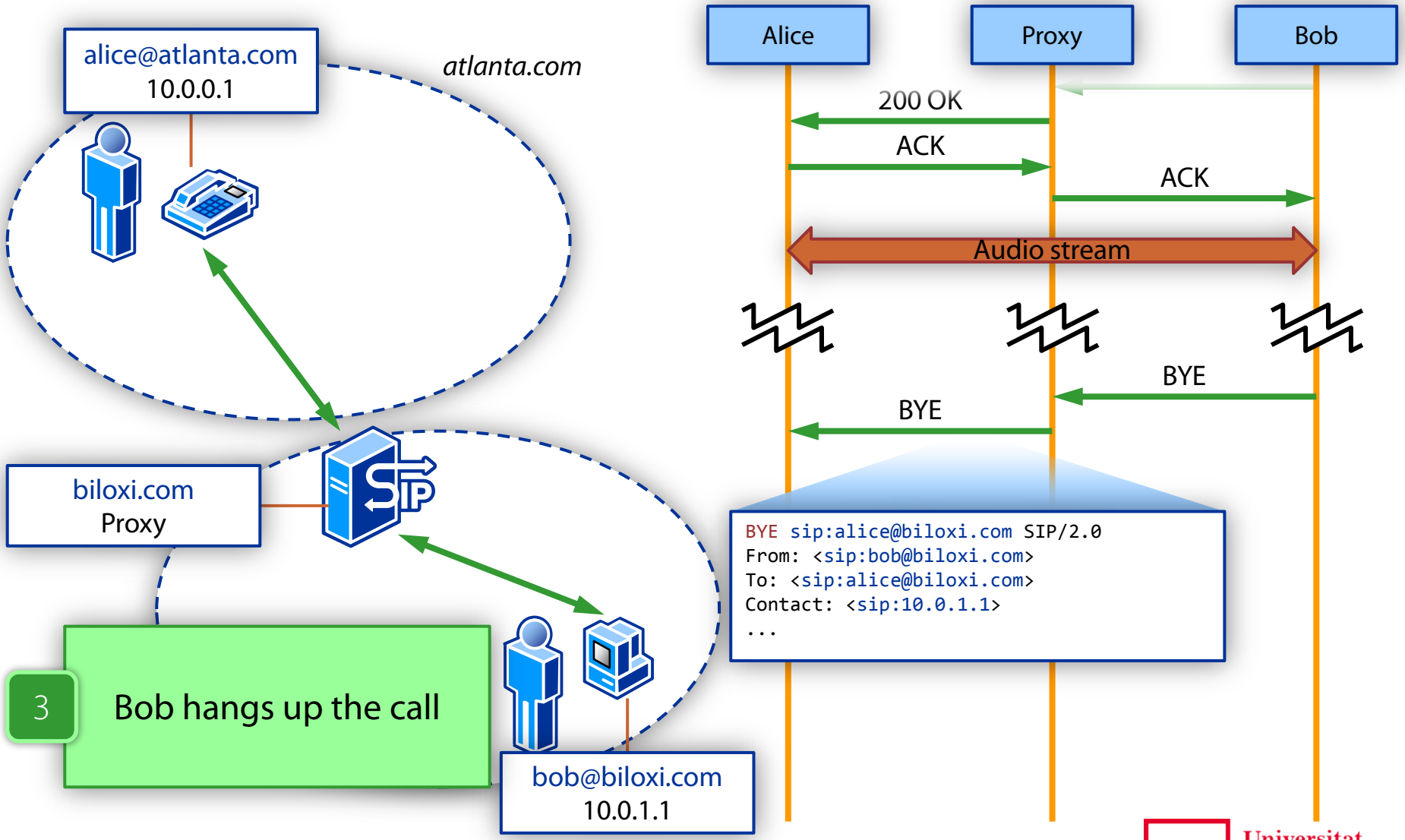
Session Establishment



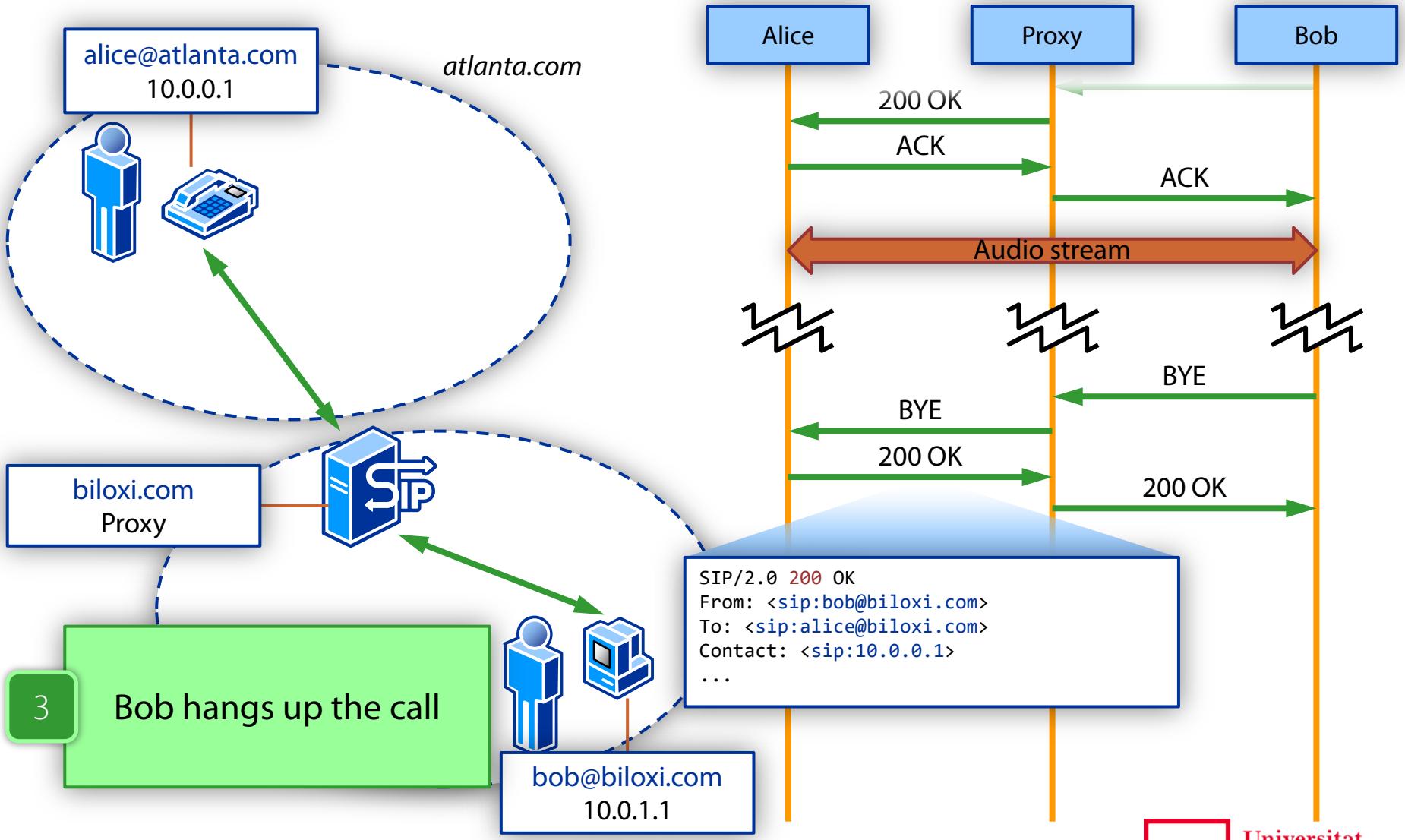
Session Establishment



Session Establishment

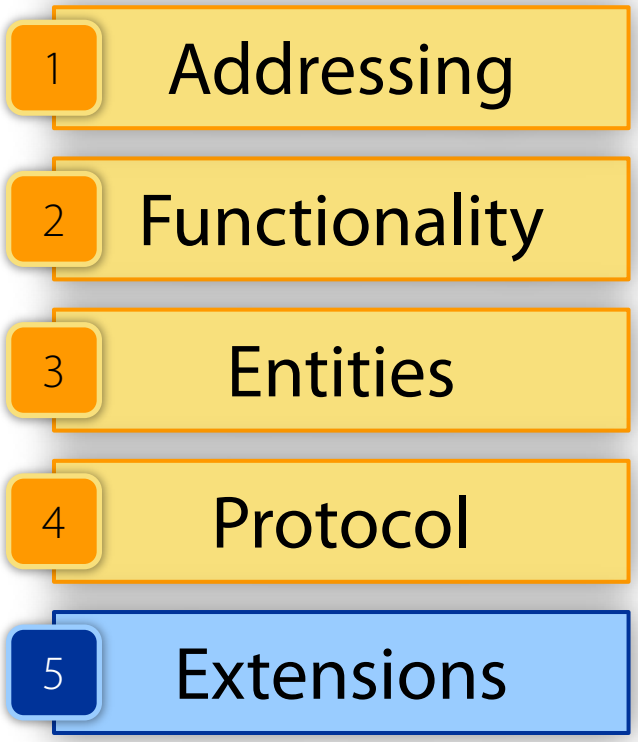


Session Establishment



Let's Recap

SIP



SIP Extensions

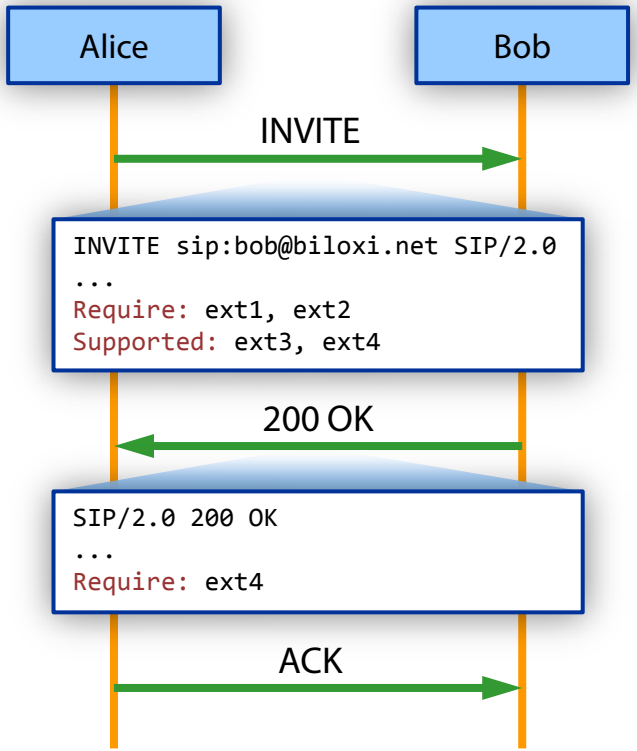
SIP features not part of the base standard

- Their implementation is optional…
- …but offer functionality beyond the core protocol

Q. How do SIP entities know which extensions are supported?

A. We have three headers

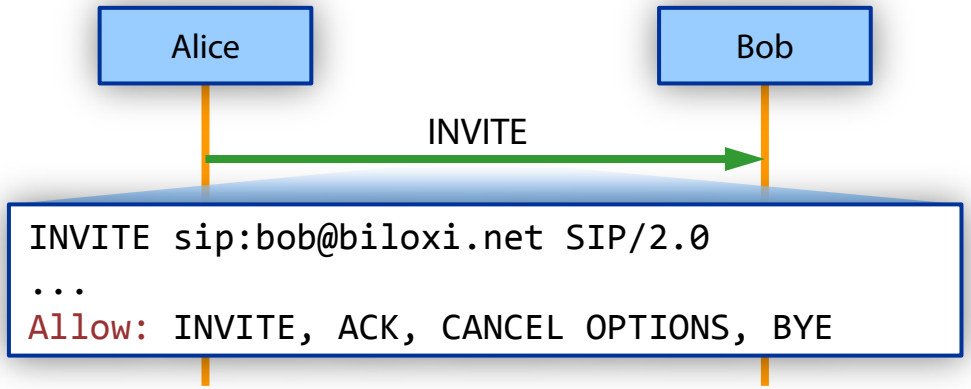
1. **Require**
   - The extensions required for a given session
2. **Supported**
   - The extensions supported by a given user agent
3. **Unsupported**
   - The extensions not supported by a given user agent



New Methods

SIP allows us to define new request methods

- An `Allow` header field indicates which methods are supported



What happens with unknown methods and headers?

Proxy	• Forwards unknown methods and headers
Redirect Server	• Ignores unknown methods and headers
UAS	• Ignores unknown headers and rejects unknown methods

Examples of SIP Extensions

RFC 3262 — Reliable Provisional Responses

- In the base protocol, provisional responses (1xx) are not transmitted reliably
- Each provisional response receives a sequence number in a new header field RSeq

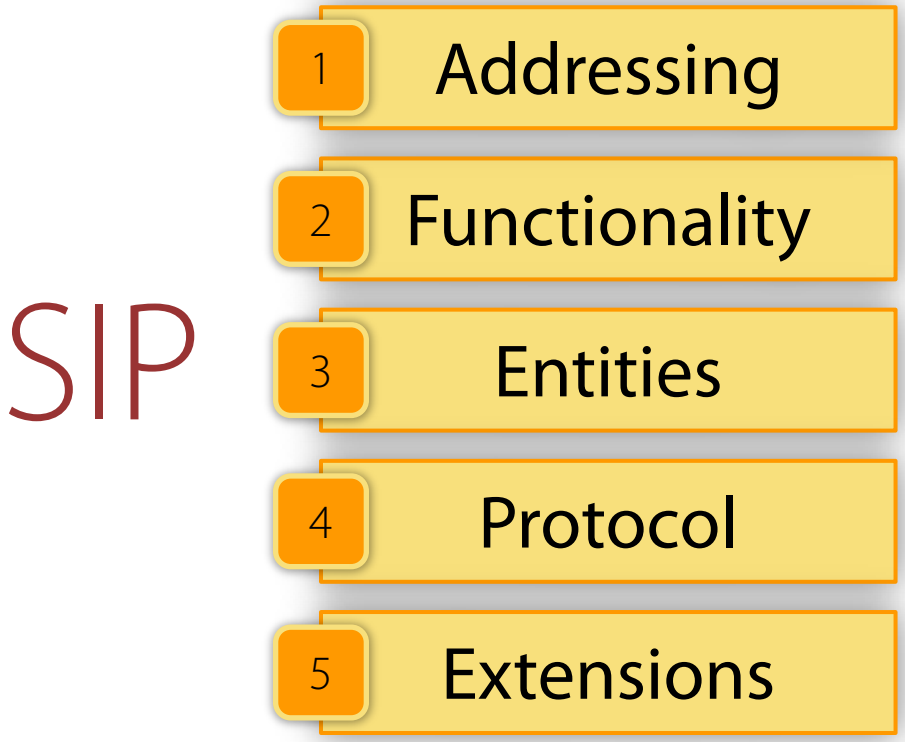
RFC 3312, RFC 4032 — Preconditions

- Constraints in the session description (SDP)
- Allows the endpoints to exchange information on when the constraints are met

RFC 3265 — Notifications

- Introduces two new methods SUBSCRIBE and NOTIFY
- SIP entities subscribe to notifications from other entities
- They receive a notification when the information or state they subscribed to changes

We're Done!



Let's Summarize

- What did we learn?

SIP

The IETF protocol to establish multimedia sessions in the Internet

1	Addressing	SIP URI				
2	Functionality	Location	Availability	Capabilities	Session Setup	Session Management
3	Entities	User Agent	Registrar	Proxy	Redirect Server	Back-to-Back User Agent
4	Protocol	Messages	Transactions	Dialogs		
5	Extensions	Negotiation	New Methods	Examples		

Multimedia in Packet Networks

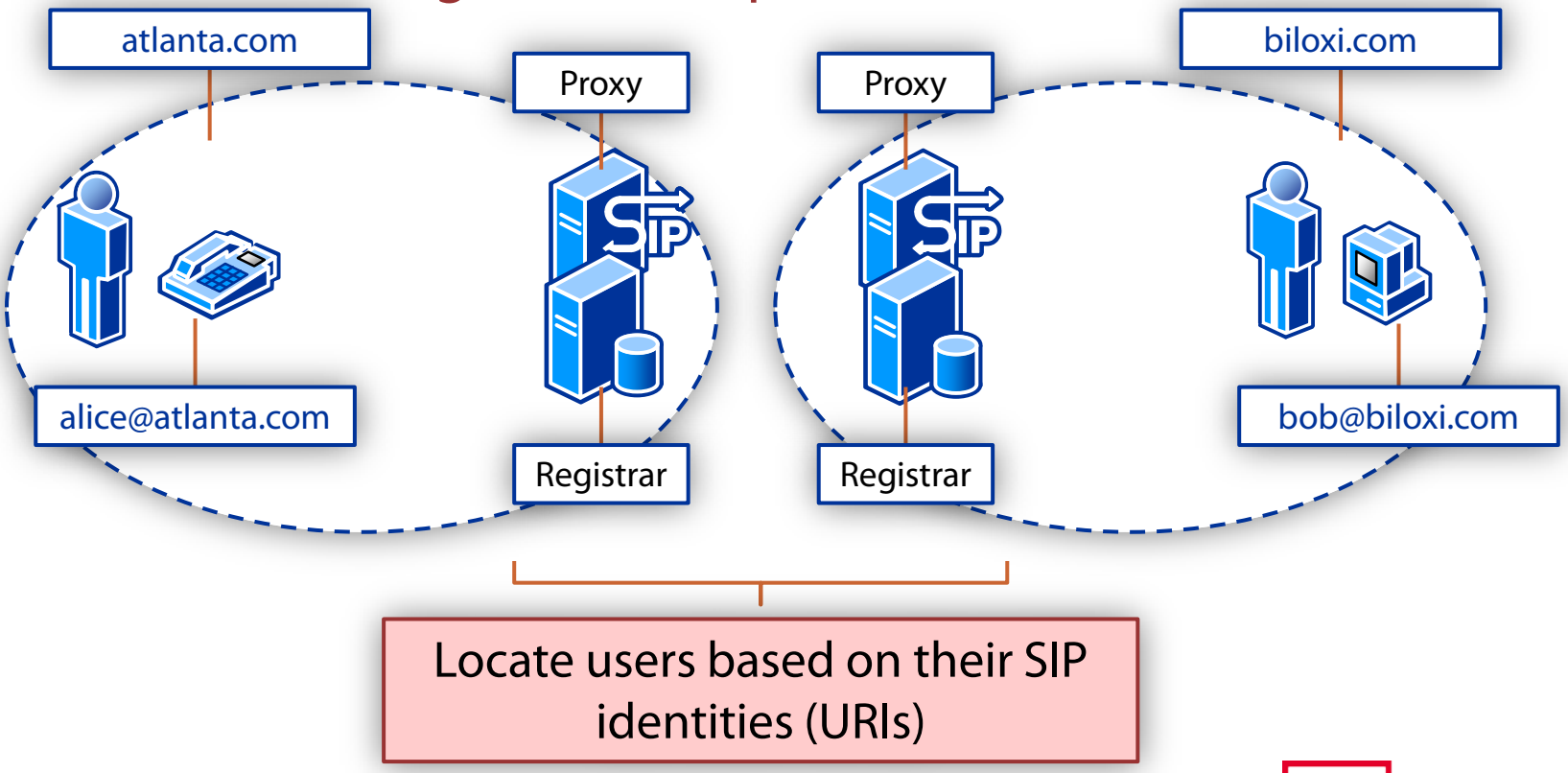
P2PSIP

Why Peer-to-Peer SIP?

SIP is a centralized protocol

Q. Why?

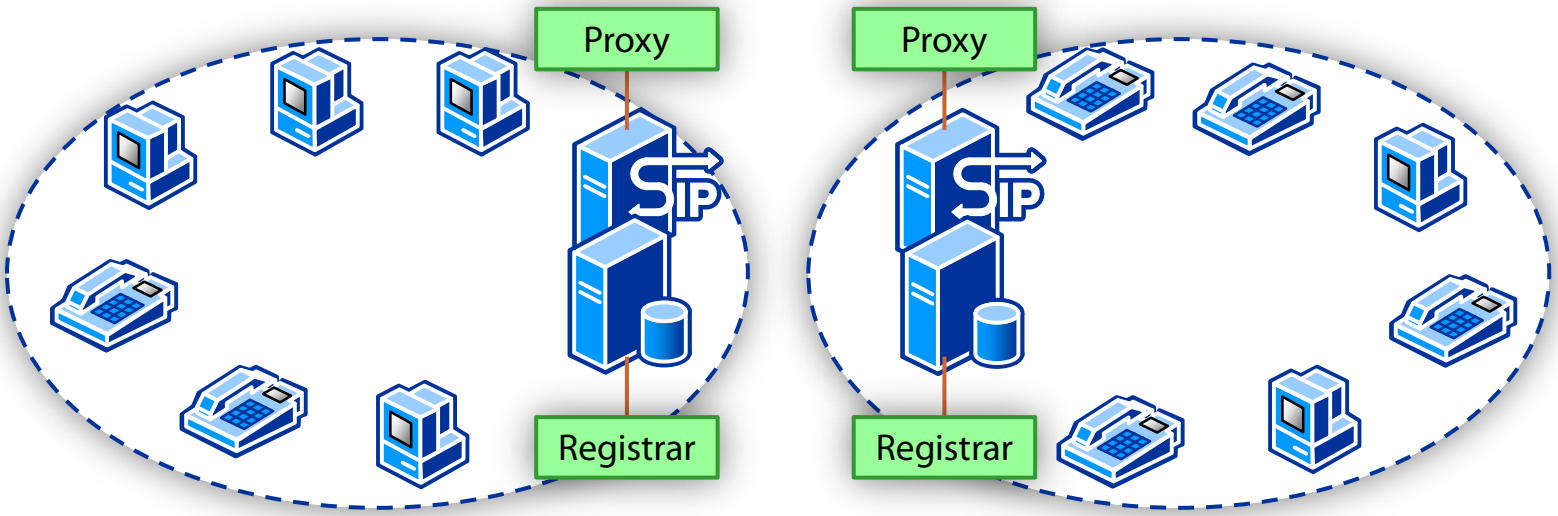
A. We need servers: registrars and proxies



Peer-to-Peer SIP

P2PSIP We get rid of the servers!

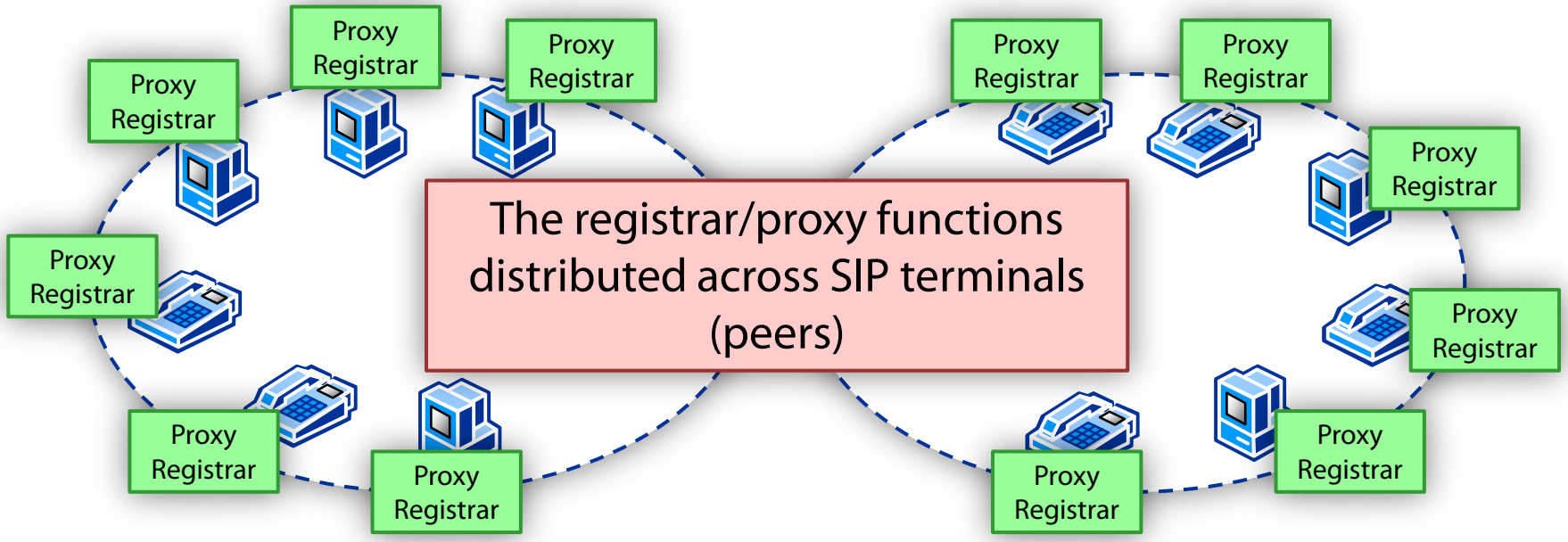
Like skype™



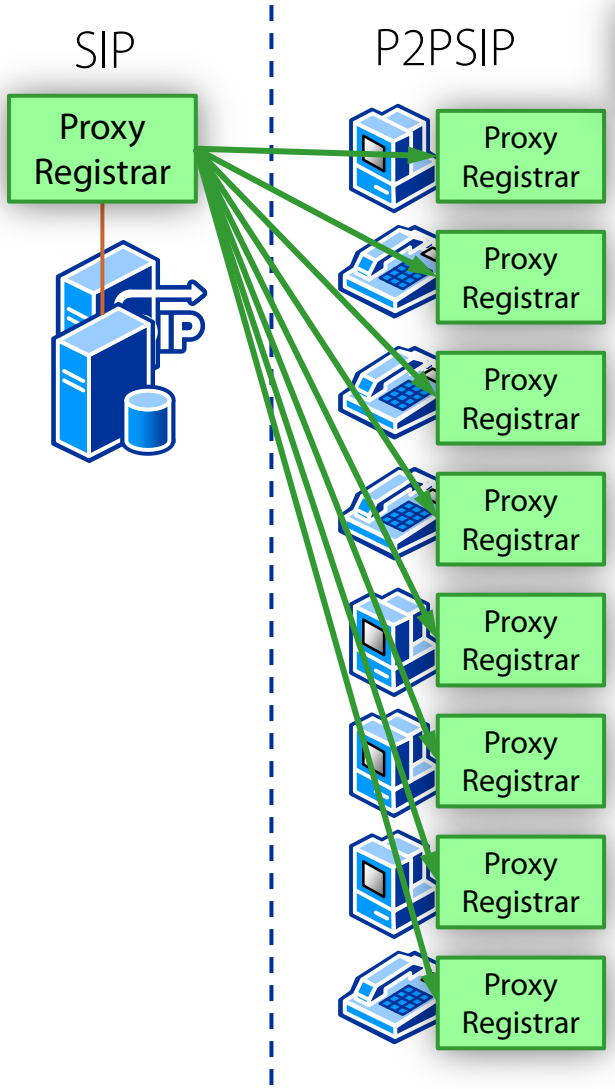
Peer-to-Peer SIP

P2PSIP We get rid of the servers!

Like skype™



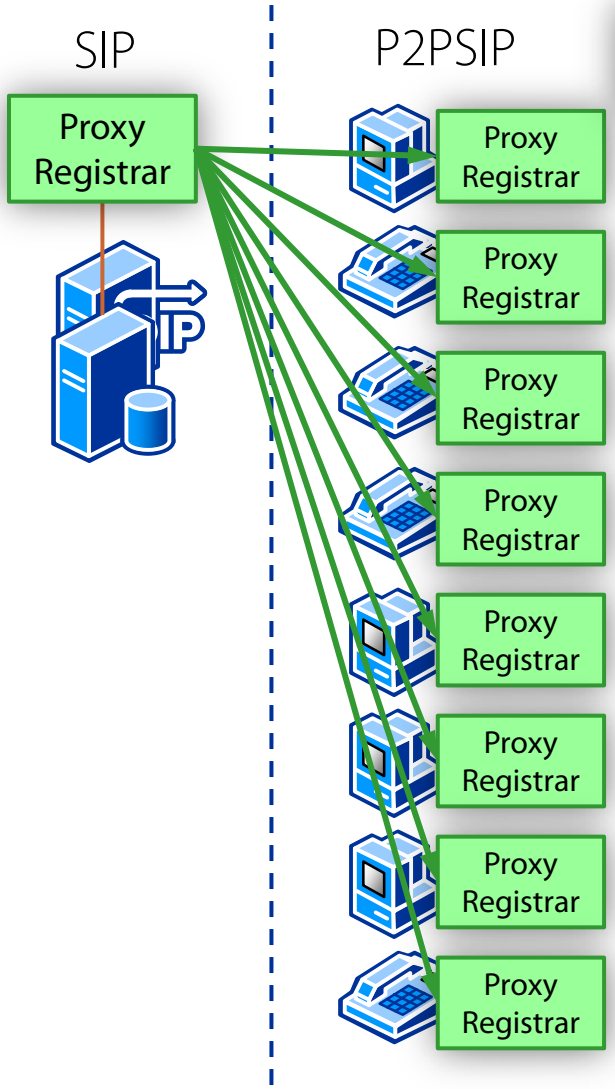
Peer-to-Peer SIP



Why P2PSIP?

- Each P2PSIP peer takes a fraction of proxy/registrar responsibility
- Lower cost (no servers)
- Scalability, the capacity (memory, storage, processing power) increases with more users
- Robustness, when a peer fails the system is still functional

Peer-to-Peer SIP



However…

- Instead of one server we have many peers
- Before, the registrar database and proxy function at one place
- Now, the registrar database and proxy function at many places
- Peers may connect, disconnect or fail at any time

P2PSIP uses a protocol called RELOAD to manage this distributed proxy/registrar

As of January 2013, at the draft status

Multimedia in Packet Networks

H.323

SIP

P2PSIP

See you next time

upf. Universitat Pompeu Fabra Barcelona