T-110.5140 Network Application Frameworks and XML

Summary and Conclusions

29.3.2010

Tancred Lindholm, Sasu Tarkoma

The lecture about everything

- Naming, addressing, routing
- Mobility
- HIP, I3, DHT, Overlays
- Middleware
- Web Services
- SOAP, UDDI, XML Signatures
- Service Federation, SSO
- Studying NAFs

The Good Olde Internet

- Goal: universal end-to-end connectivity
- Multiplexing
  - Packet switching
- Survivability (robustness)
  - Dynamic adaptation to outages
- Service generality
  - Support widest possible set of applications
- Runs over diverse networking technologies
  - Heterogeneity is unavoidable

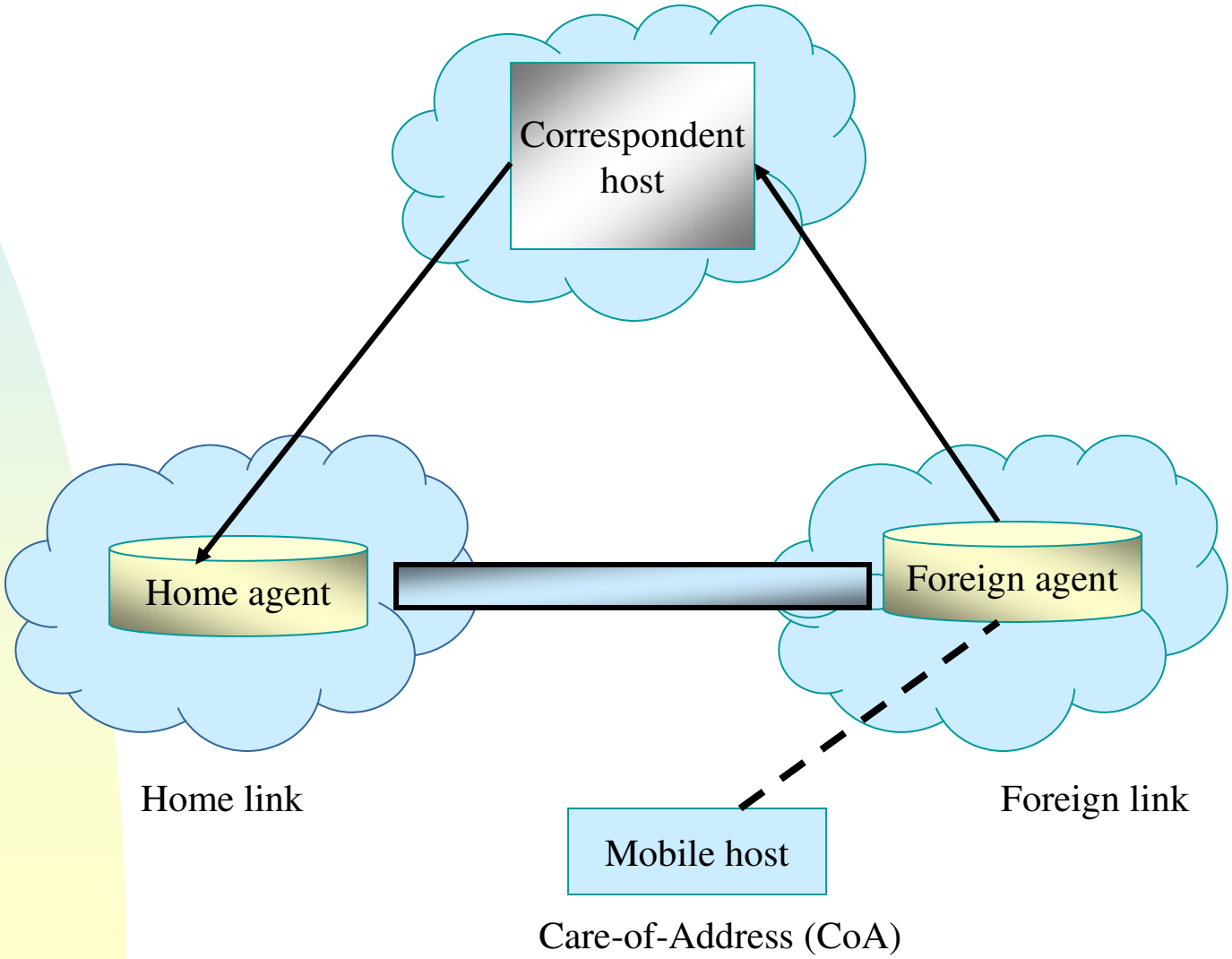
What has changed?

- Permanent IP address
  - Time-varying: DHCP, NAT, mobility
- End-to-end communication
  - Middleboxes, proxies, NATs, ..
- Globally and uniquely routable
  - NAT, firewalls
- Trusted end hosts
  - Hackers, spammers, …
- Four layers
  - Layer splits, cross-layer interactions

Routing vs. Mobility

- Topology data aggregation is necessary
  - Cannot track all hosts in the world
  - IP addresses determined by topology
    - Network gives the routing prefix
- Mobile hosts must change their IP addresses
  - Causes sockets / connections to break
- How to communicate address changes?
- Goal of a mobility protocol
  - Transport and applications do not see address changes
  - Mobility transparency

Mobility Example: Mobile IP Triangular Routing



The Identifier/Locator Problem

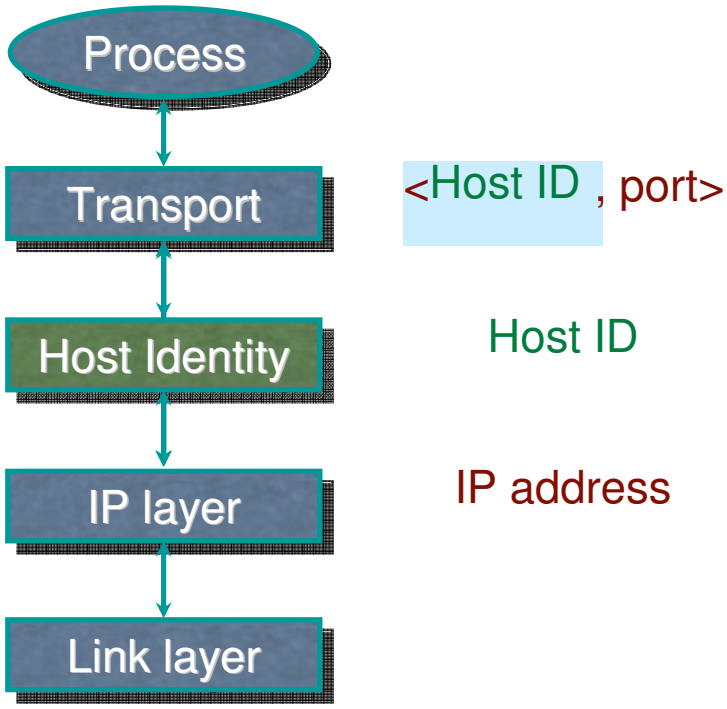
- Problem: machine.domain.com is both name **and** address (b/c DNS limitations, early resolution to IP address)
- New name space for entity IDs
  - Maybe based on DNS?
  - Maybe a separate namespace?
  - Maybe IP addresses are used for location?
- Communication end-points (sockets) bound to identifiers, not addresses

HIP: Splitting the locator from identity

- HIP = Host Identity Protocol
- A proposal to separate identifier from locator at the network layer of the TCP/IP stack
  - A new name space of public keys
  - A protocol for discovering and authenticating bindings between public keys and IP addresses
- Secured using signatures and keyed hashes (hash in combination with a secret key)

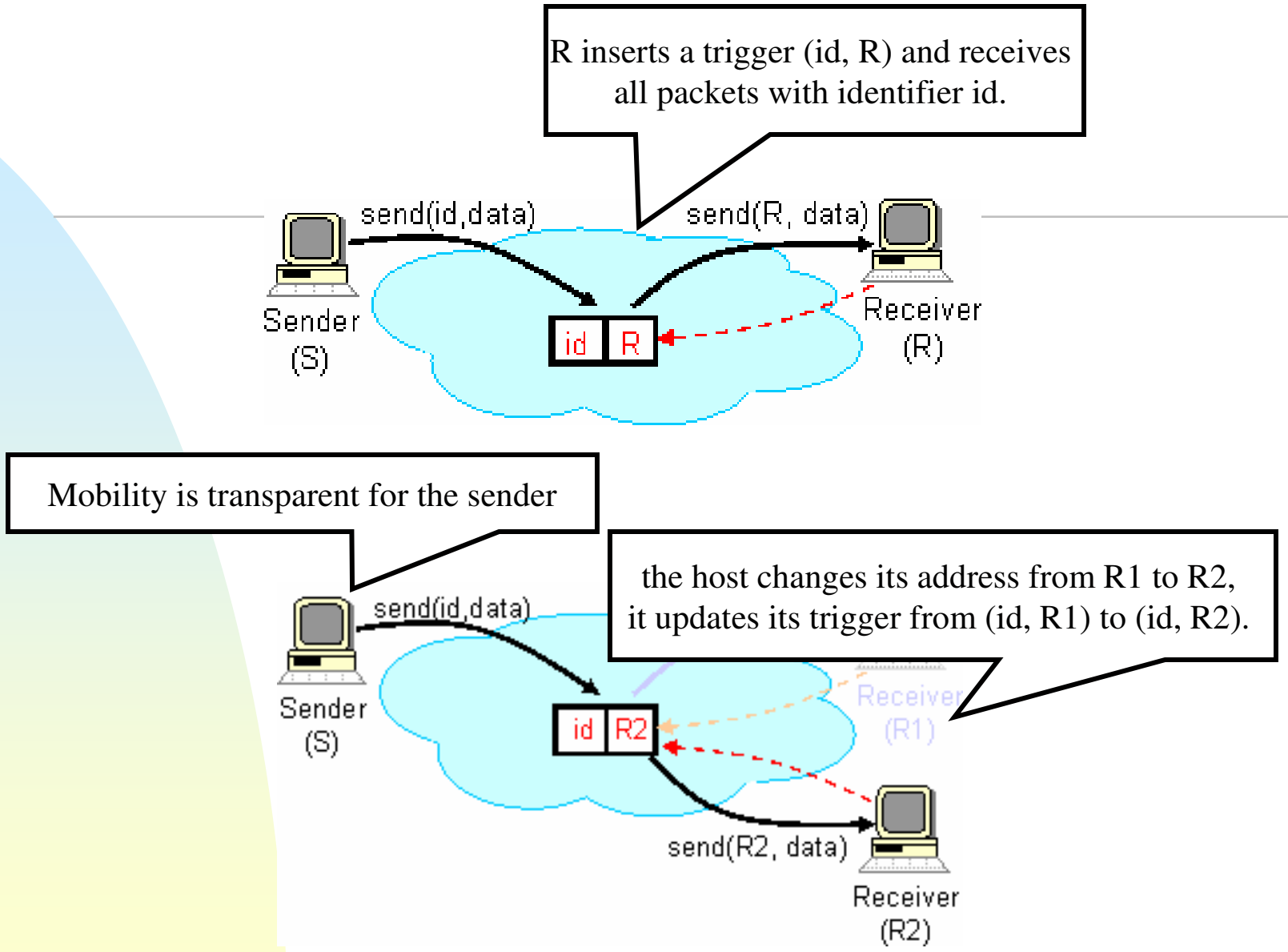
The Idea

- A new Name Space of Host Identifiers (HI)
- Public crypto keys!
- Presented as 128-bit long hash values,
  Host ID Tags (HIT)
- Sockets bound to HIs, not to IP addresses
- HIs translated to IP addresses in the kernel



Internet Indirection Infrastructure (i3)

- A DHT - based overlay network
  - Based on Chord
- Aims to provide more flexible communication model than current IP addressing
- Decouples sender from receiver by introducing indirection point
- One proposal to fix some fundamental problems in the Internet

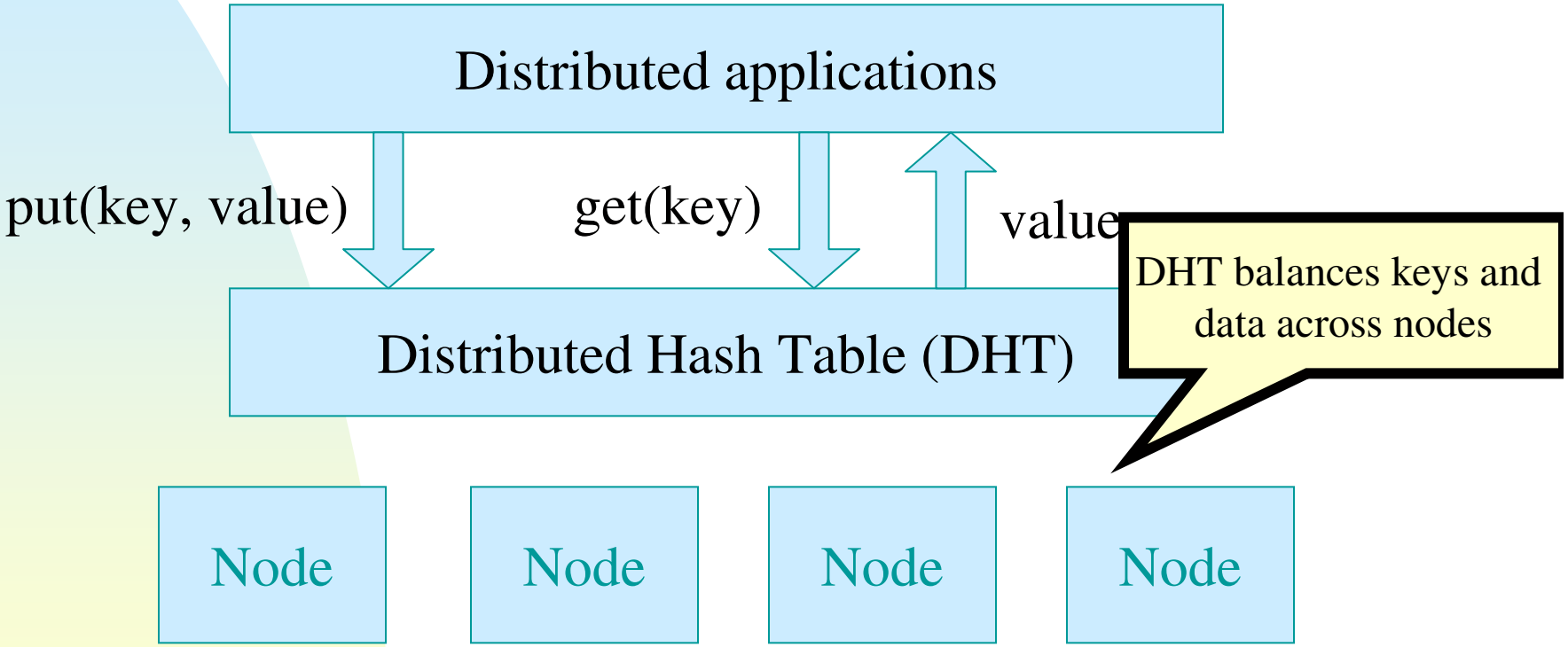


Source: http://i3.cs.berkeley.edu/

DHT Motivation

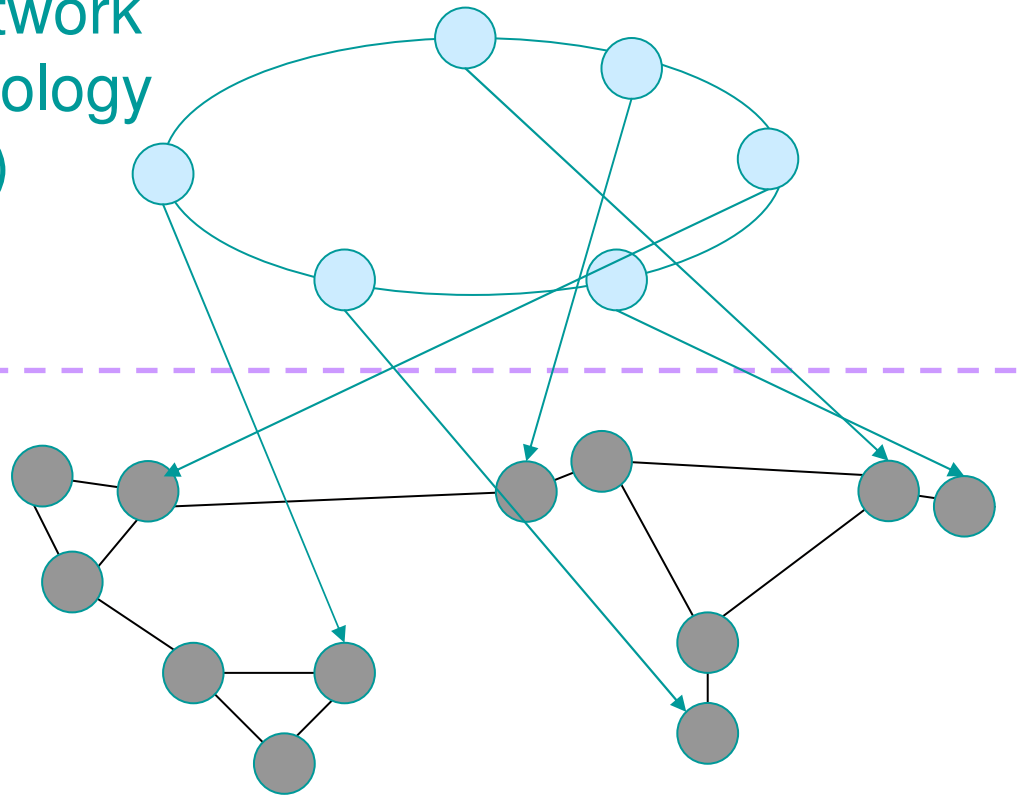
- Directories are needed
  - Name resolution & lookup
  - Mobility support with fast updates
- Required properties
  - Fast updates
  - Scalability
  - Reliability

DHT Operations



Building Overlay Networks with DHT

Overlay Network
Logical Topology
(e.g. Chord)



Node "real" topology in IP network

Middleware

- Application development is complex and time-consuming
  - Should every developer code their own protocols for directories, transactions, ..?
  - How to cope with heterogeneous environments?
- Middleware is needed
  - To cut down development time
    - Rapid application development
  - Simplify the development of applications
  - Support heterogeneous environments and mask differences in OS/languages/hardware

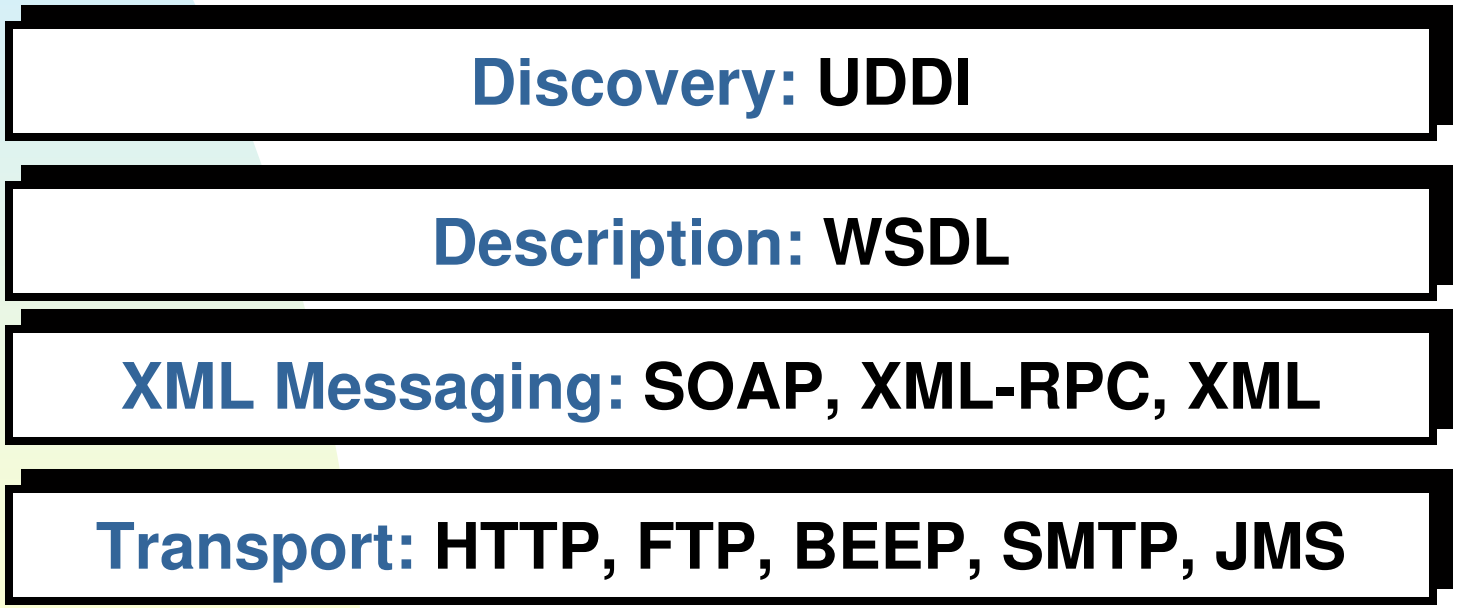
Middleware Examples

- DHTs
- Event Systems
  - some nodes publish data on topics
  - other nodes subscribe on interesting topics
  - asynchronous model
  - event queues
  - Example: Java Messaging Service
- Web Services

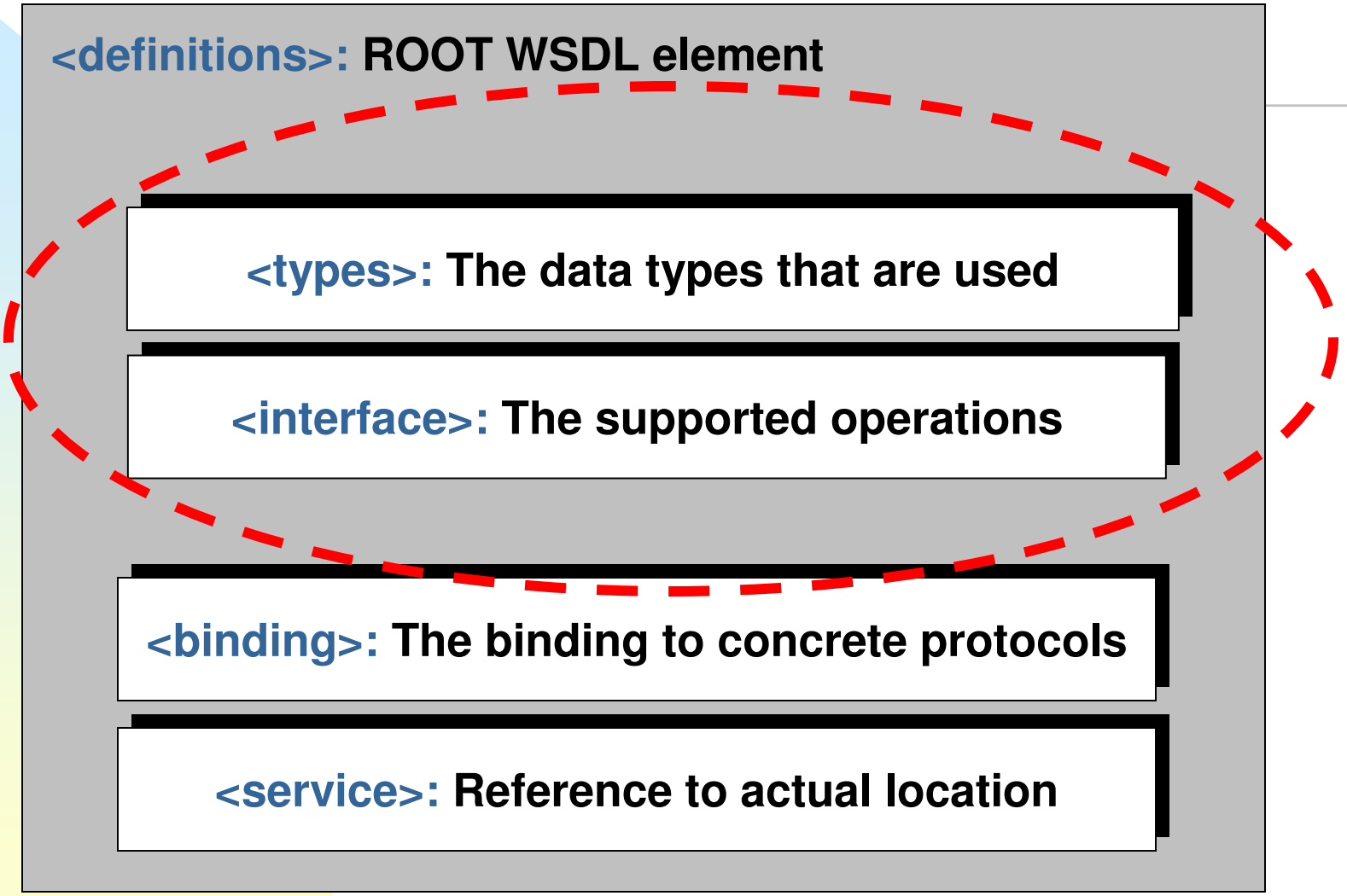
Web Services

- Let's make machine-callable services using web principles
- A central role is played by the description of the service's interface
- Implementation less important, avoid implementation-specifics
- Business aspects considered
  - Use across organizations
  - Multiple competing implementations

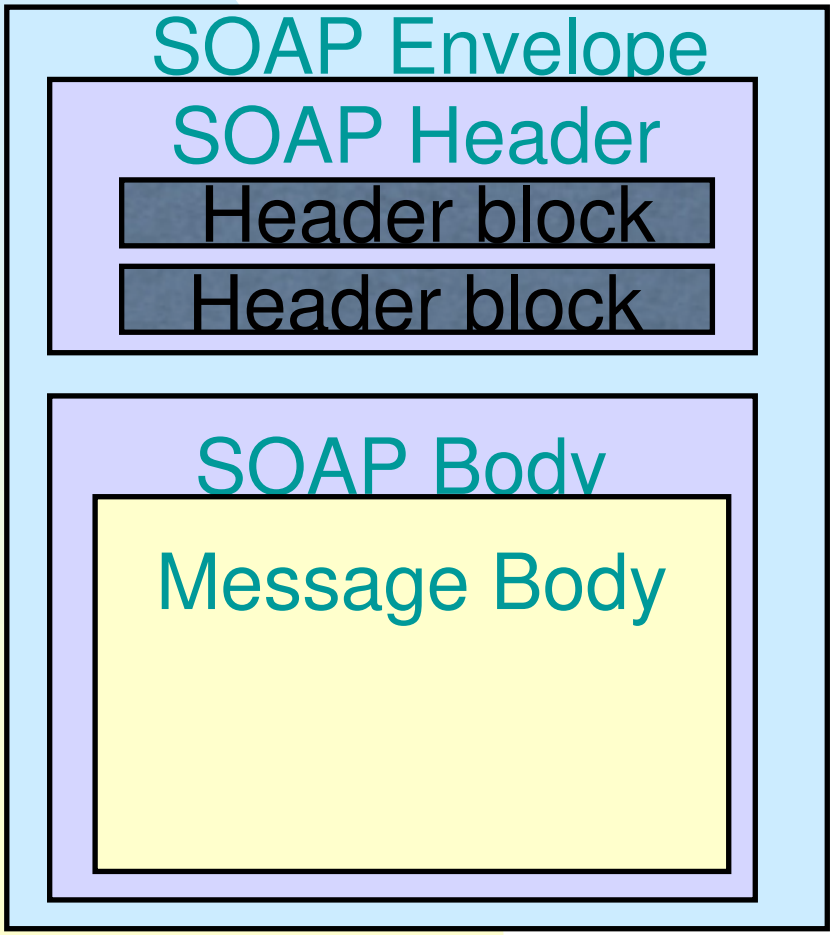
WS Protocol Stack



WSDL Overview



SOAP Message Structure



- Optional header contains blocks of information regarding how to process the message:
  - Routing and delivery settings
  - Authentication/authorization assertions
  - Transaction contexts
- Body is a mandatory element and contains the actual message to be delivered and processed (and fault information)

RPC/encoded-style SOAP Message

```
public Float getQuote(String symbol);

<s:Envelope
 xmlns:s=http://www.w3.org/2001/06/soap-envelope>
  <s:Header>
    <m:transaction xmlns:m="soap-transaction"
            s:mustUnderstand="true">
      <transactionID>1234</transactionID>
    </m:transaction>
  </s:Header>
  <s:Body>
    <n:getQuote xmlns:n="http://example/QuoteService.wsdl">
      <symbol xsi:type="xsd:string">IBM</symbol>
    </n:getQuote>
  </s:Body>
</s:Envelope>
```

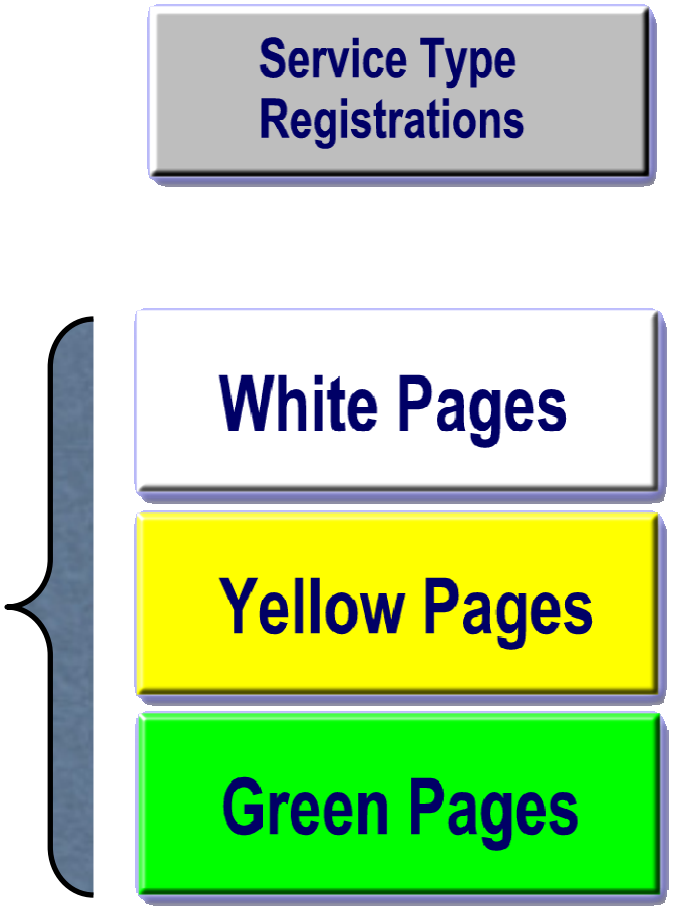
UDDI

- Universal Description Discovery and Integration
- A “meta service” for locating web services by enabling robust queries against rich metadata
- Distributed registry of businesses and their service descriptions implemented in a common XML format

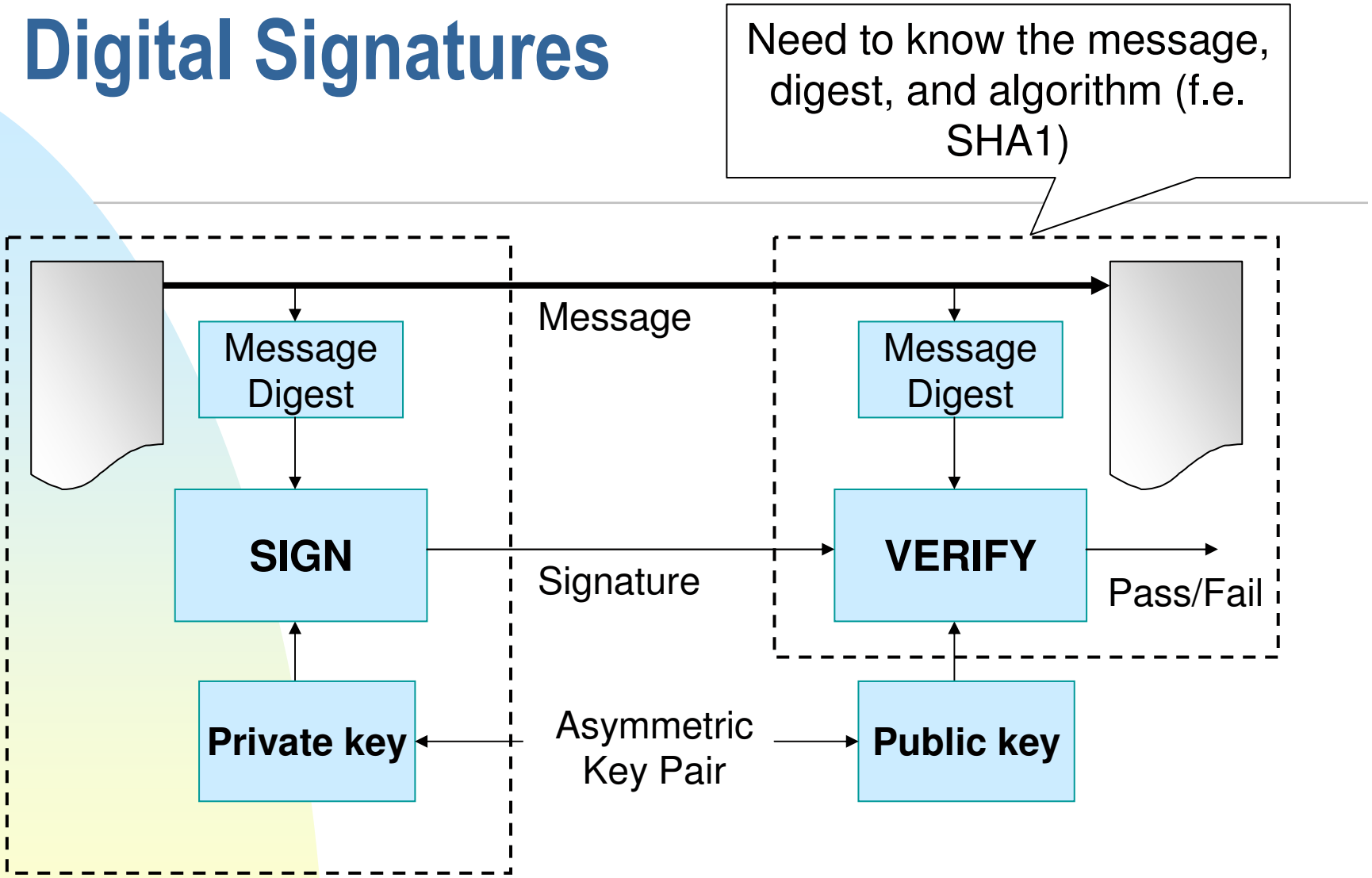
UDDI Registry Entries

Standards Bodies, Agencies, Programmers, Publishers register specifications for their Service Types

Service providers register precise information about themselves and their Web services



Digital Signatures



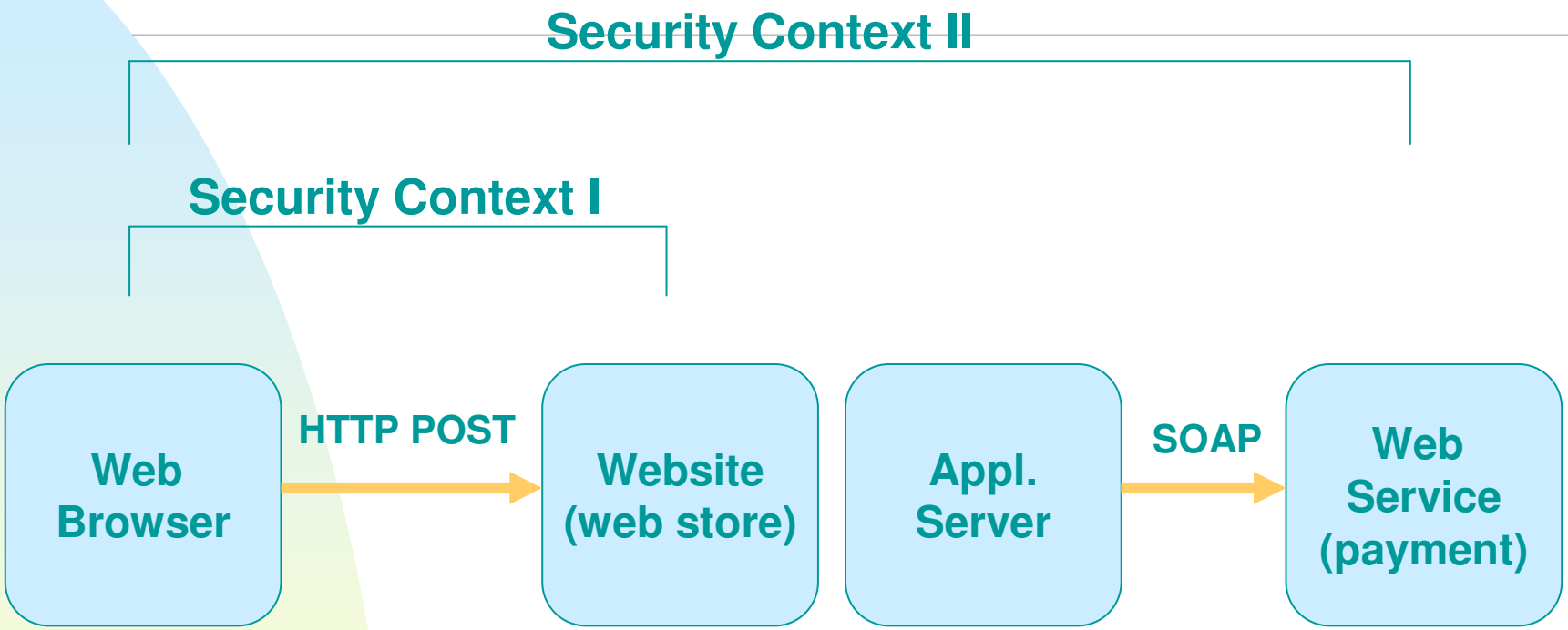
Need for XML security

- XML document can be encrypted using SSL or IPSec
  - this cannot handle the different parts of the document
  - documents may be routed hop-by-hop
  - different entities must process different parts of the document
- SSL/TLS/IPSec provide message integrity and privacy only when the message is in transit
- We also need to encrypt and authenticate the document in arbitrary sequences and to involve multiple parties

Security Contexts Across Web Services

- Remember Web Services goals:
  - Re-use existing services
  - Combine services from several domains
- Security result: Must support several security domains
  - SOAP intermediaries
  - Reusing security tokens from one message in another message

Example: Passing sensitive information



Main Point: We need security within AND between security contexts!

SAML for exhanging security assertions

- SAML (Security Assertion Markup Language)
- XML-based framework for exchanging security information
  - XML-encoded security assertions
  - XML-encoded request/response protocol
  - Rules on using assertions with standard transport and messaging frameworks
- Example: Authetication
  - An issuing authority asserts that:
    - Subject S
    - was authenticated by means M
    - at time T

Single sign-on (SSO)

- Most important (?) use case for multi-party security assertion
- OpenID is a popular Web SSO

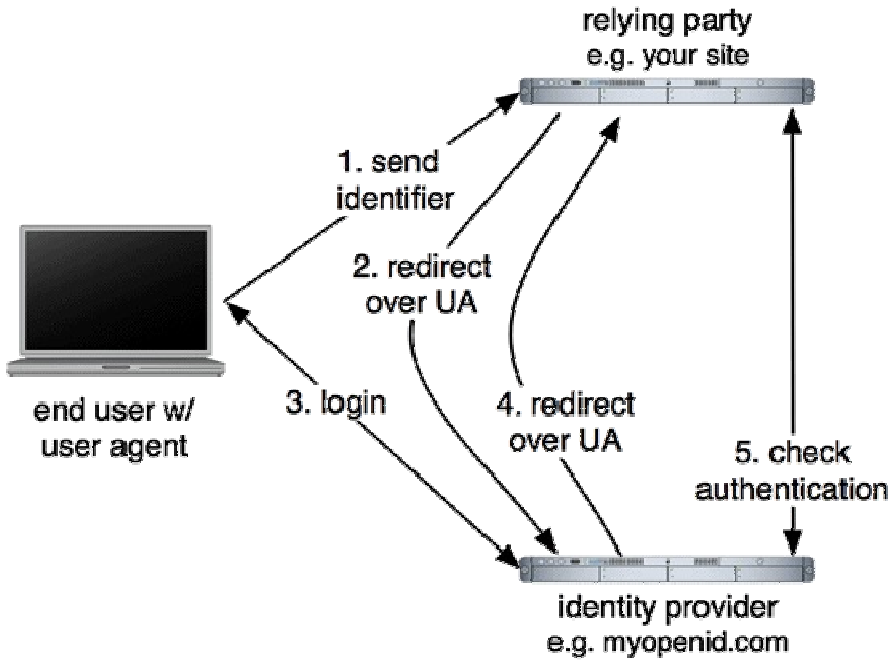
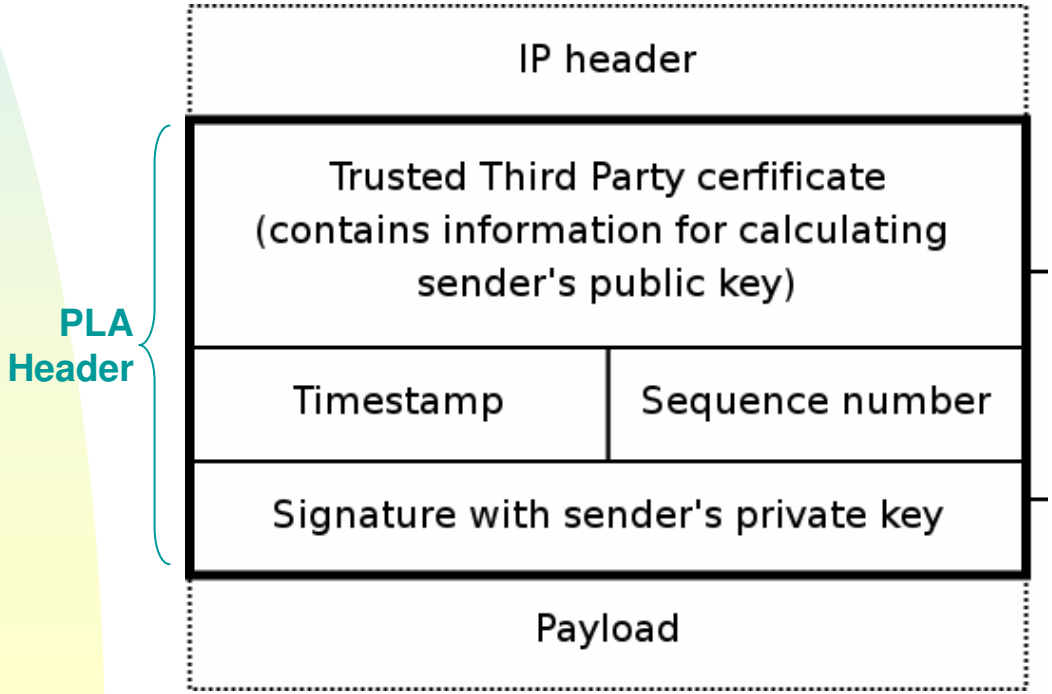


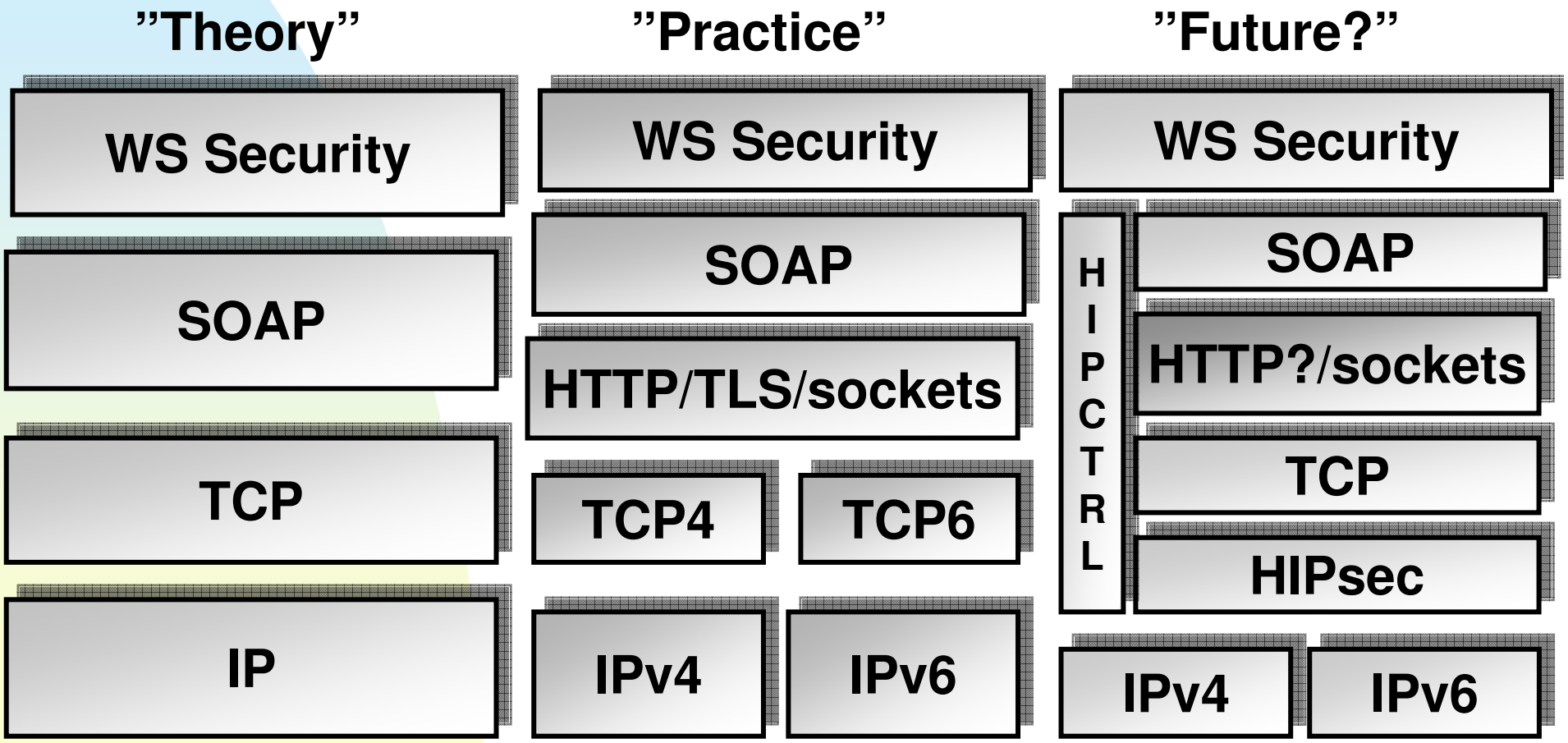
Image from http://farm1.static.flickr.com/163/351494842_cd83fef2f5_o.jpg

Packet Level Authentication (PLA)

- Per packet signatures
- Any node can verify authenticity of every packet without previous trust



Putting it all together



Studying NAFs

- We learned a few NAFs, 100s remain (present and future)
- Goal: ability to quickly understand further NAFs
  1. Know the fundaments (signatures, data exchange patterns, object models, ...)
  2. Identify the key concepts/abstractions
  3. See through implementation details
  4. Be critical – trying to pick something apart is a good way to learn!

Thank you!

- Questions?