FFI Forsvarets forskningsinstitutt

Identity Management Systems for multi-domain and tactical networks

Finseskolen 2012

Anders Fongen, PhD
Norwegian Defence Research Establishment

May 2012

Why Identity Management (IdM)?

- Because we need integrated management of info about
  - Users
  - Equipment
  - Roles and Privileges
- A PKI (Public Key Infrastructure) only manages keys
- **IdM = Identity Provision+Authentication+Access Control**

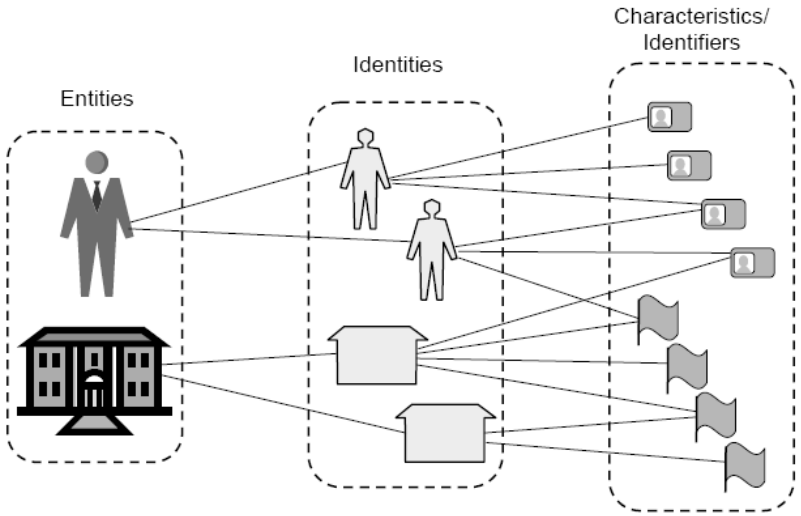


Figure 1: Correspondence between entities, identities and characteristics/identifiers.

Identity Management (IdM)

- Identity:
  - Set of properties associated with an *Entity*
- Identifier:
  - Subset of properties to *distinguish* identities
- Identity Statement:
  - Attestation of the subject's identifier
- Identity Provider (IdP)
  - Service which issues identity statements
- Authentication
  - Establishment of identity

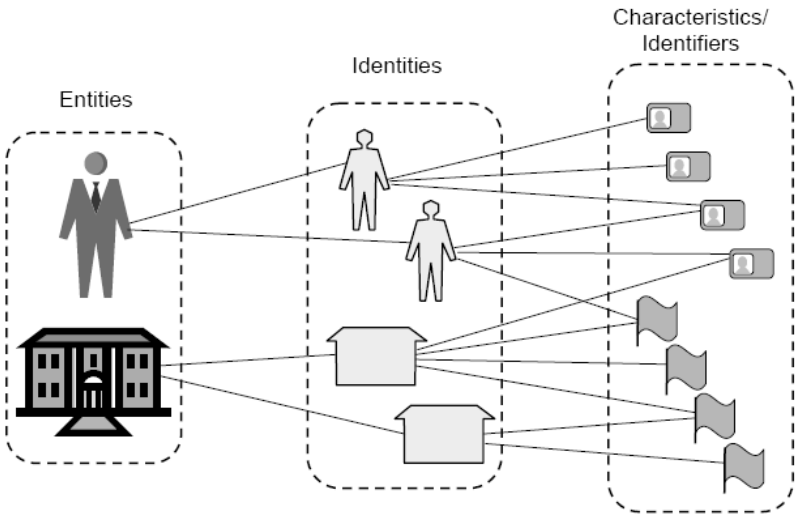


Figure 1: Correspondence between entities, identities and characteristics/identifiers.

IdM for mobile/tactical systems

- Are likely targets for intrusion attacks
- Highly dynamic client population
  - Even coalition networks of “guest” client
- Lightweight client units
  - Portability issues
    - Library availability
    - SOAP support
- Communication network
  - Availability/connectivity
  - Latency (round-trip delay)
  - Bandwidth (communication delay)

IdMs are suffering from:

- Discarding existing investments
  - need separate user registries
- High coupling between domains
  - guest users individually registered
  - autonomy delegated for federation
- Visibility of user identities
  - access given to identities, not roles
- Driven by security excellence, not networking excellence
  - protocols too costly for ”narrow and bumpy” networks

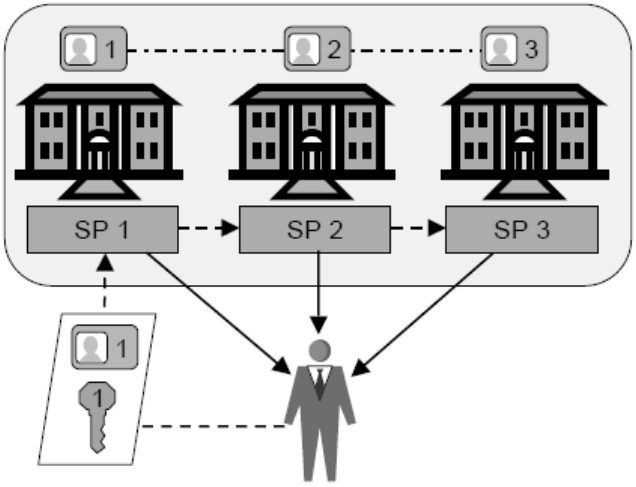


Figure 3: Federated user identity model.

IdM systems should

- Protect investments and knowledge
  - Employ existing enrollment procedures and data storage
- Allow federation for ”guest access”
  - Should not need to enroll guests
- Give access rights to *roles*, not identities
  - RBAC, ABAC
- Protect domain autonomy
  - owner of service decides the access control
- Allow system latency
  - trust has a lifetime
- Limit the trust relationships
  - minimize the ”trust anchors”
- Balance requirements between security and network economy

Tactical networks – ubiquitous computing

- Mobile, wireless, based on military radio technology
  - spread spectrum, strong encryption
- Low bandwidth (< 100 kb/s, depending on range)
- Multi-hop, Ad-hoc
  - latency
  - packet loss
  - link loss

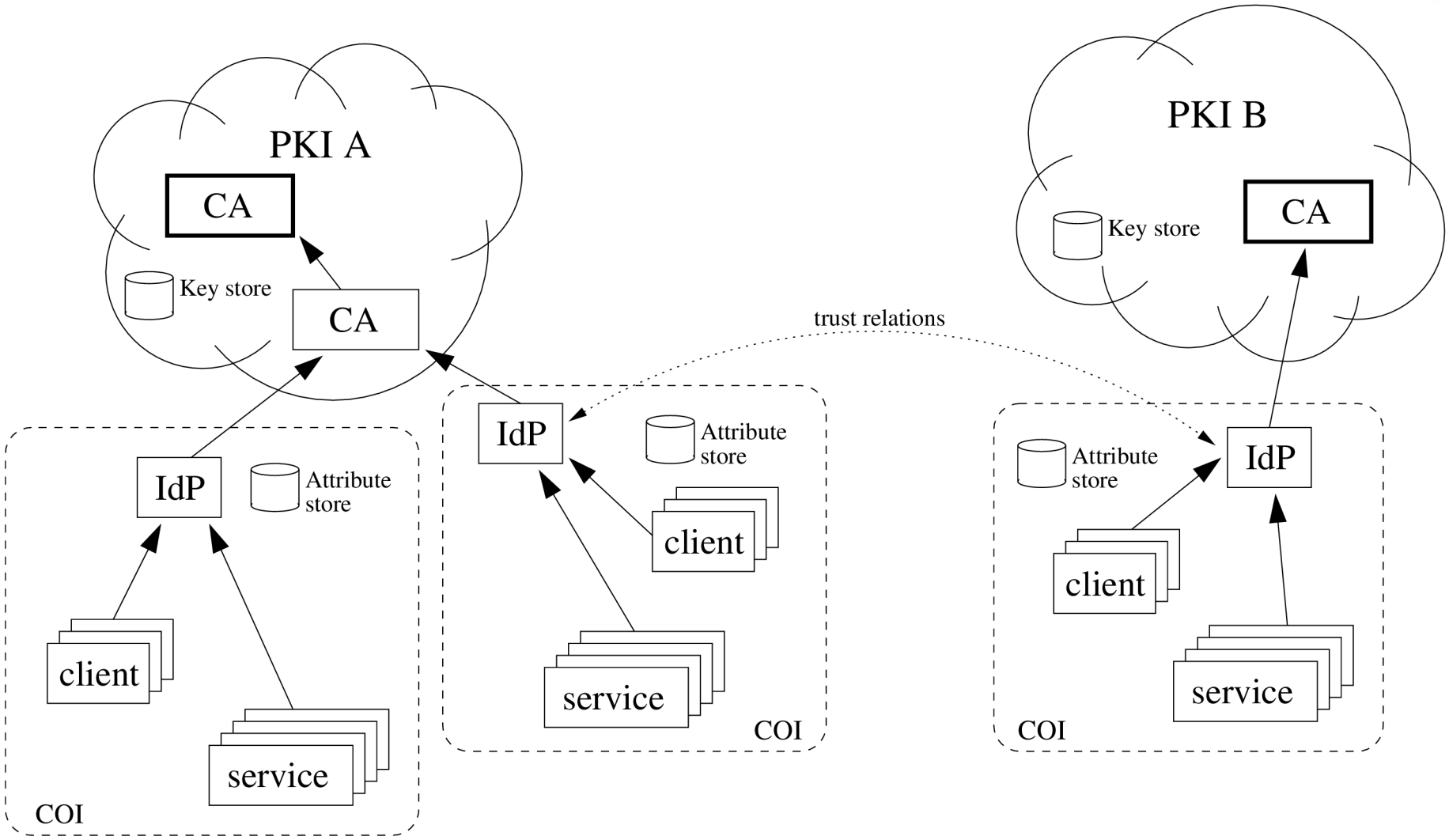
- Applications adapted for tactical networks are frugal, robust and perserverant, which are desirable properties everywhere
  - ***tactical applications are fit for ubiquitous computing***

Revocation and Tactical Networks

- Identity credentials may need to be *revoked*
- Revocation of identity information requires bandwidth and connectivity
- Revocation checking is expensive and error-prone
  - since one actually asks the *opposite question*

- The work presented
  - relies on short-lived ***”identity statements”*** which require no revocation scheme,
  - the identity statements are derived from X.509 certificates maintained in a PKI

The GISMO IdM Architecture

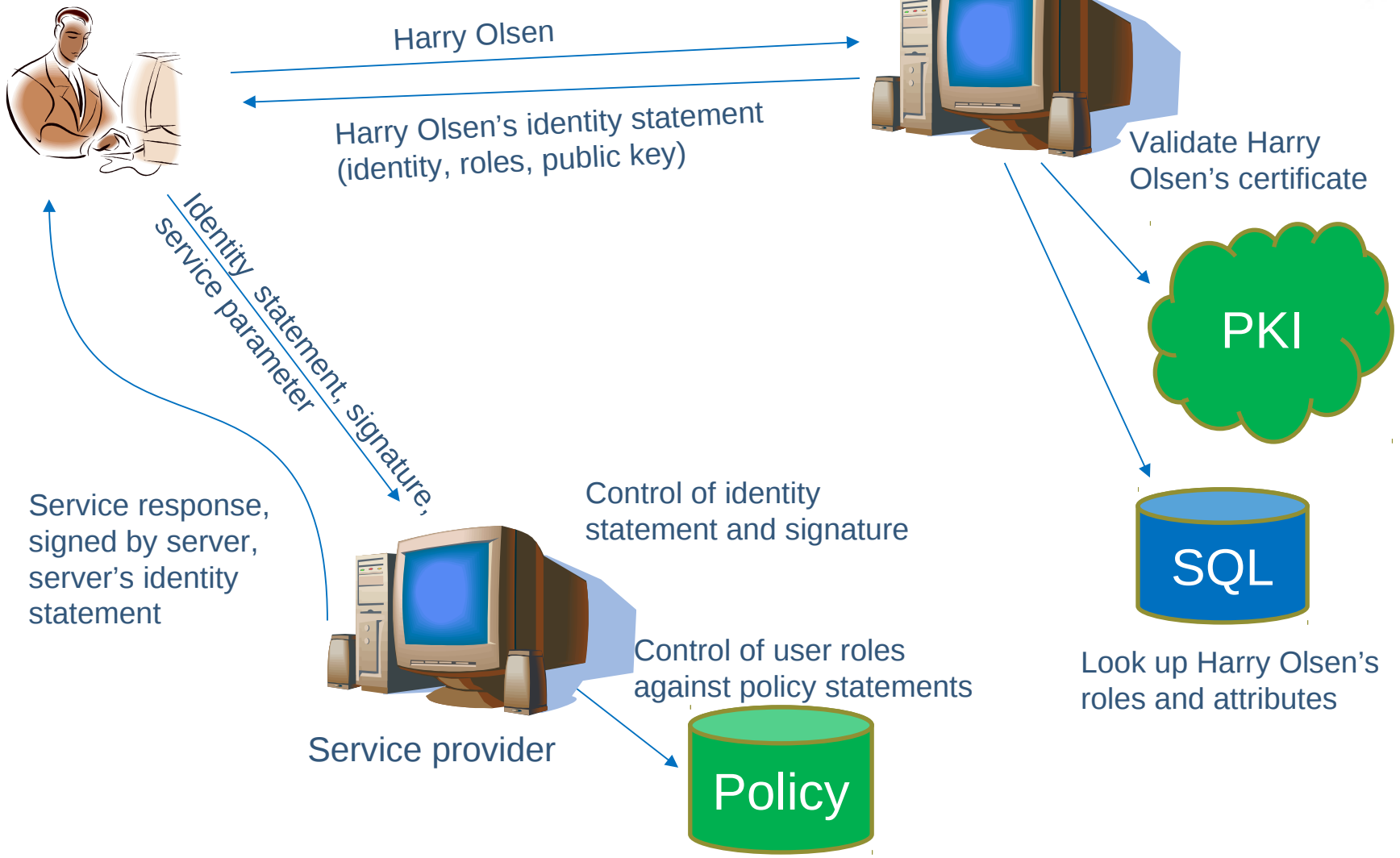


The Identity Statement

- Attested binding between properties and identifier
  - public key, attributes
- Signed by a trusted issuer
- Expires
- Both clients and services presents their identity statements in order to provide *mutual* authentication

Subject identifier
Subject public key/x509 cert
Subject attribute 1..n
Validity period (from-to)
Issuer identifier
Issuer public key/x509 cert
Issuer’s signature

Identity establishment example 1



Trust assumptions

- The identity statement is issued (and signed) by the IdM
  - The service providers need trust in the IdM
    - that the identity statement are ”correct”
- The service providers trust the authenticity of a client who demonstrates a private key (proof-of-possession)

Cross domain IdM principles:

- Inside each domain:
  - User key/certificate management
  - User roles/privileges management
- Between domains:
  - Trust in others' authentication process
  - Trust in integrity of user attributes
  - No management of foreign users
- Role based authorization process
  - since identity of guests are "unmanaged" in host domain

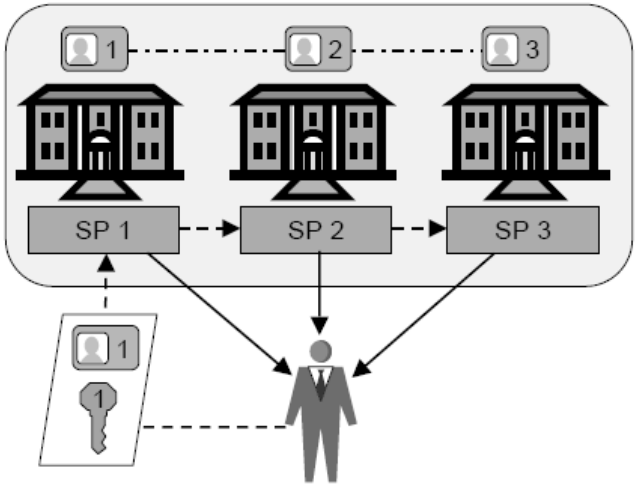
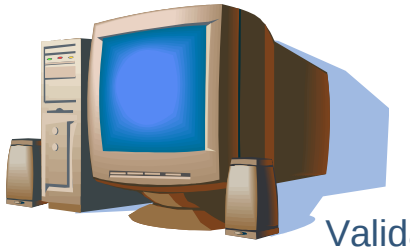


Figure 3: Federated user identity model.

Identity establishment example 2

Client A

Identity manager A



Harry Olsen

Harry Olsen's identity statement
(identity, roles, public key)

Validate Harry Olsen's certificate

Harry Olsen's identity statement

Guest identity statement

Guest identity statement, signature, service parameter

Identity manager B

PKI

Control of identity statement and signature

SQL

Look up Harry Olsen's roles and attributes

Service response, signed by server, server's identity statement

Service provider B

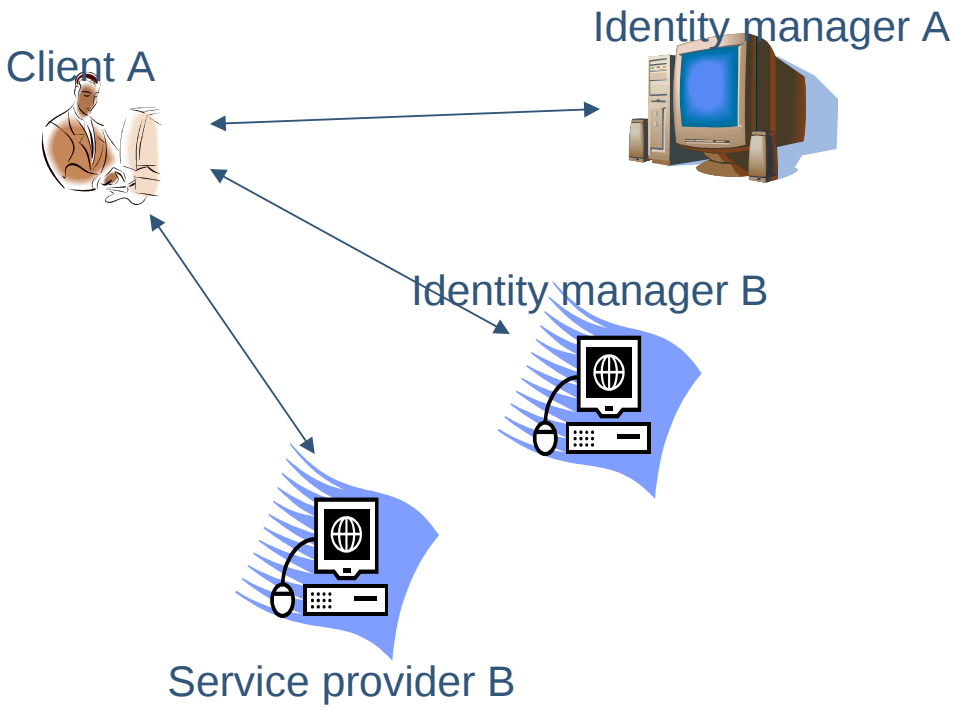
Control of identity statement and signature

Control of user roles against policy statements

Policy

Trust relations

- IdM-B (Identity manager in domain B) trusts the authentication process of IdM-A.
  - it vouches for IdM-A by *re-signing* the identity statement
    - makes it into a domain-B security document



Caching is King

- The Identity Statement can be used for all servers in the domain for the duration of its lifetime
- Reduces traffic and connectivity requirements to the Identity Provider
  - Good for mobile systems

GISMO IdM’s advantages

- Administrative and Authority Issues
  - autonomy of domains and COI
  - loose coupling between domains (certificate pair)
- Scalability issues
  - no CRL distribution
  - single domain user management
- Mobility / Tactical issues
  - occational service invocations with IdM
  - client-A and server-B can connect independent on IdM reachability

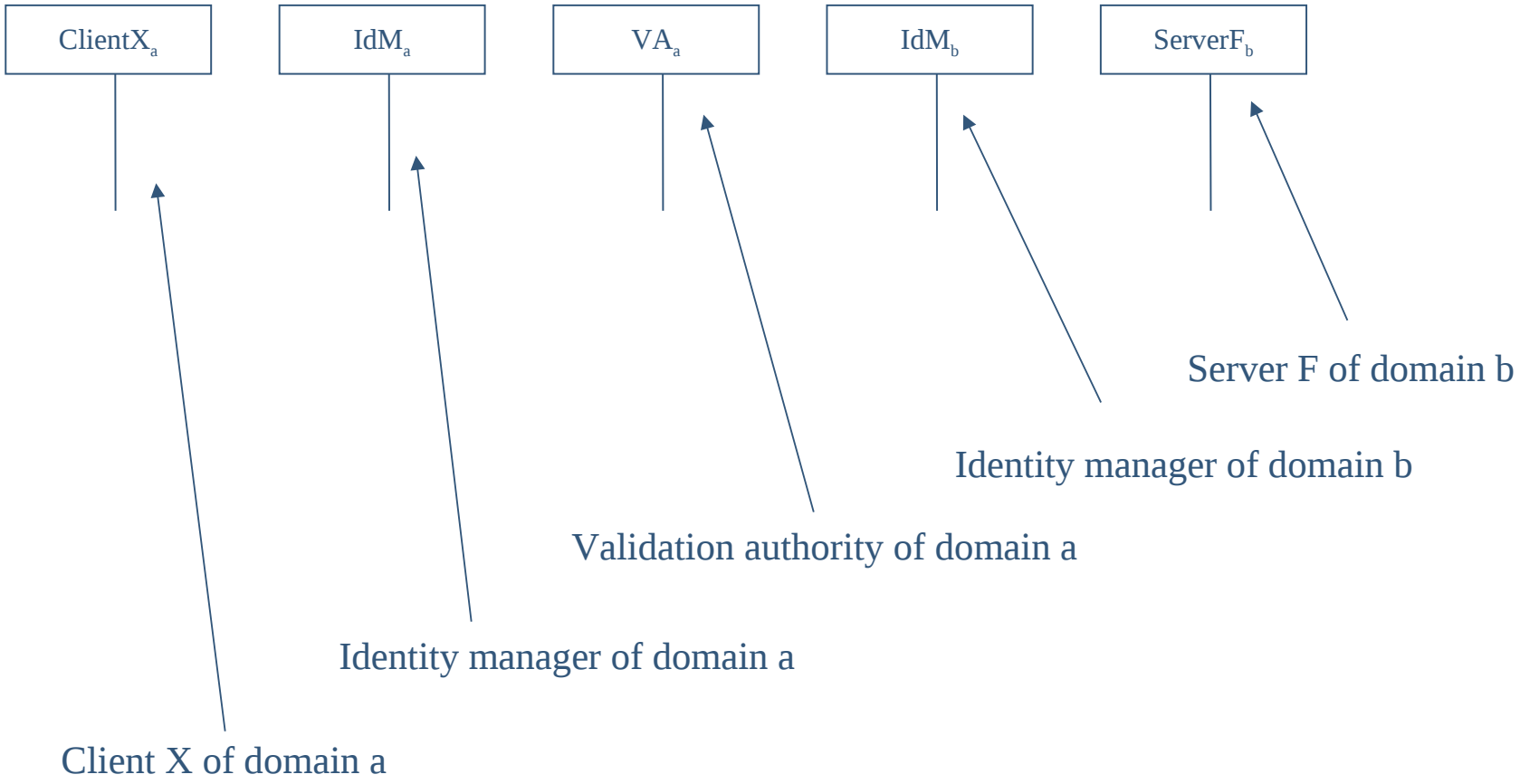
Requirements for Cross Domain IdM:

- Several security domains
- Low network traffic
- Few extra network invocations involved
- Uncomplicated and verifyable
- Suitable security level
- Application transparency

IS Issue and Authentication details

- In the following pages, a more detailed description of protocols and data structures is given.

The actors are:

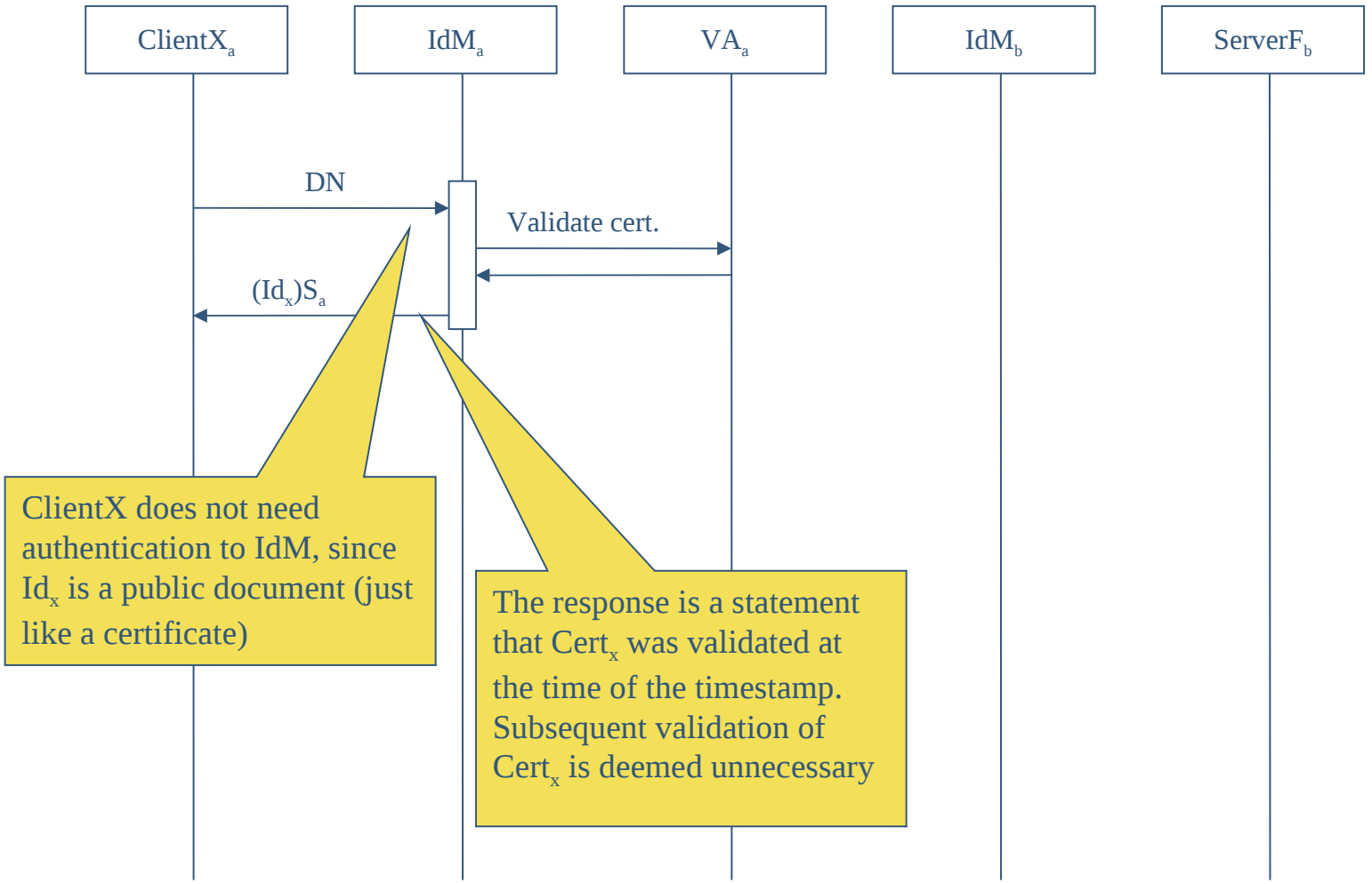


The data elements are:

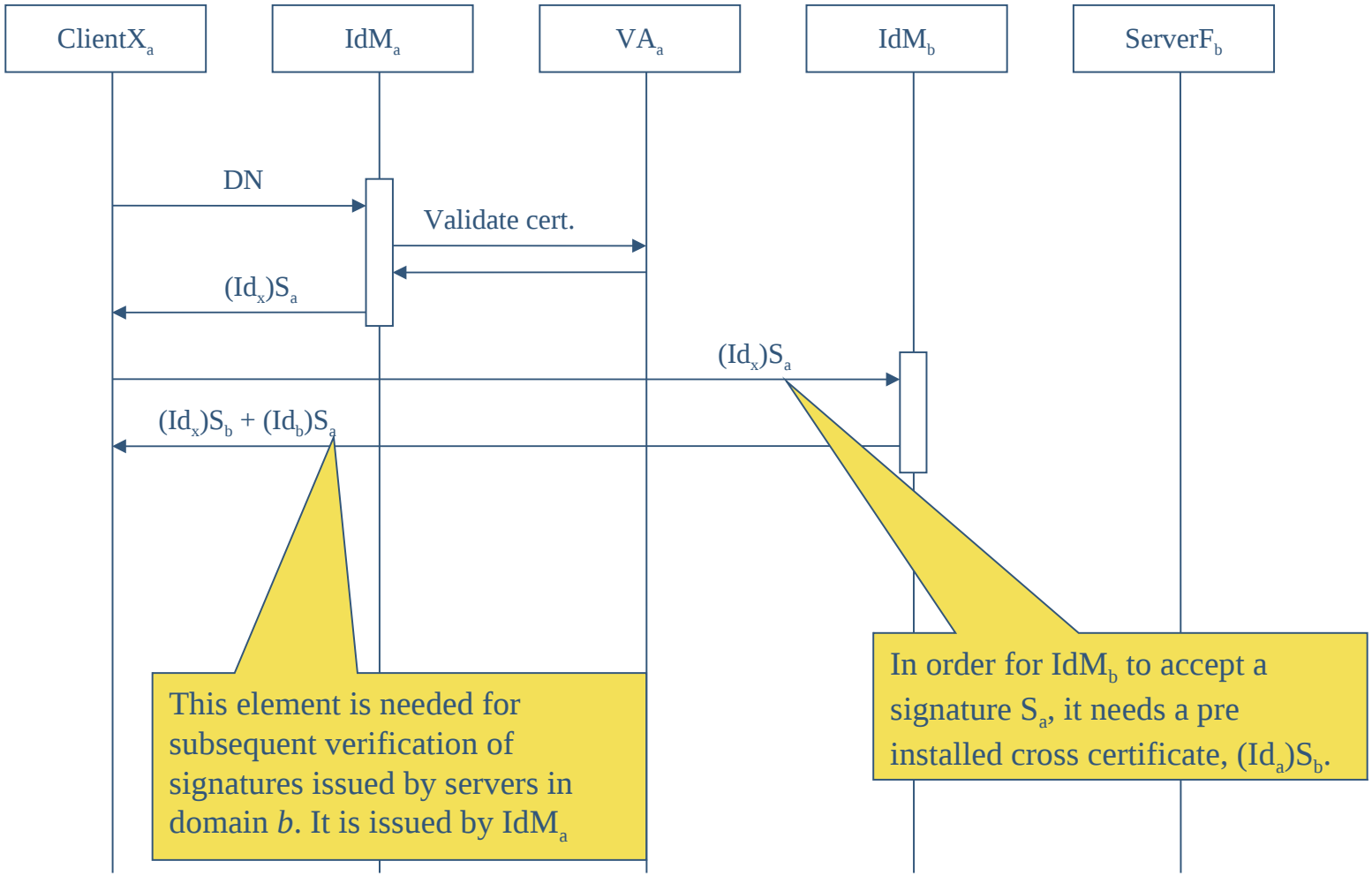
- DN = X.500 Distinguished Name (CN=Anders Fongen, O=FFI..)
- Id = X509Cert + Timestamp + Attributes
- $(Id_x)S_a$ = Id of x signed by IdM_a
- $(Message)E_x$ = Message encrypted with public key of x
- $(Message)S_x$ = Message signed with private key of x

- Keep in mind:
  - Members of domain *a* (clients and servers) have trust in the signature of IdM_a
  - No revocation info is distributed, but handled inside the VA
  - The validity period of an Id is given by the timestamp

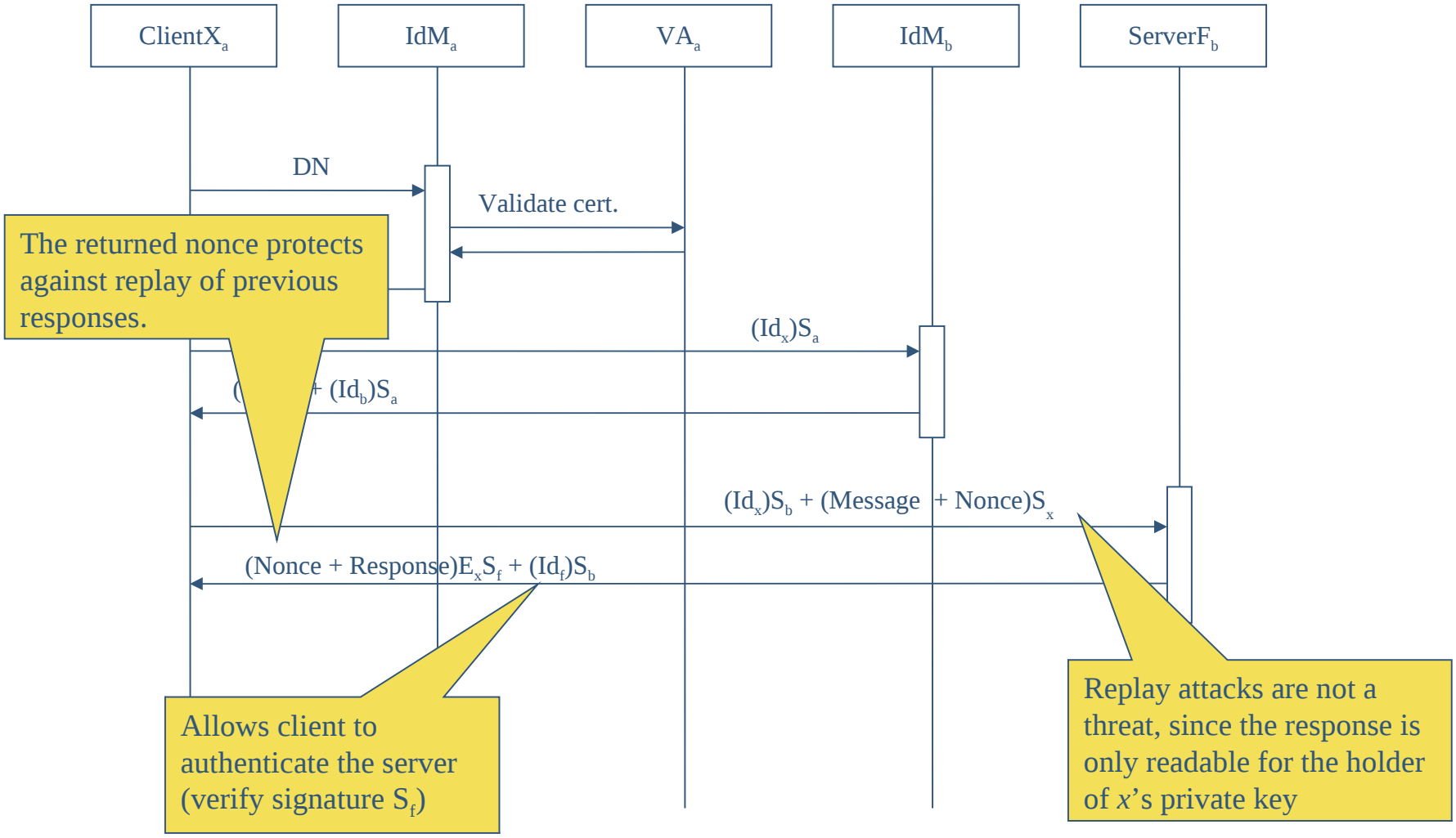
Step 1: Issue a signed Id (own domain)



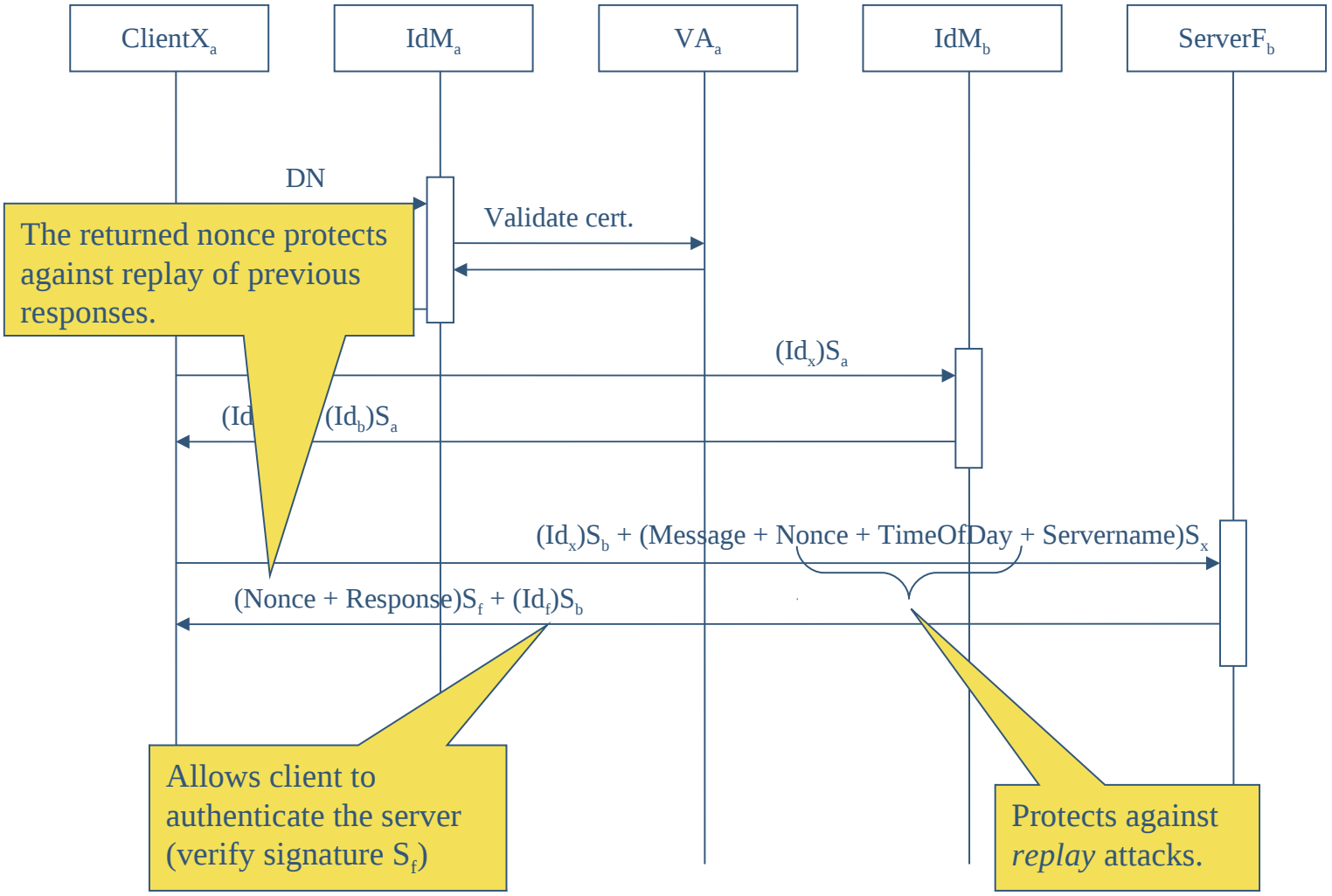
Step 2: Issue a guest Id (foreign domain)



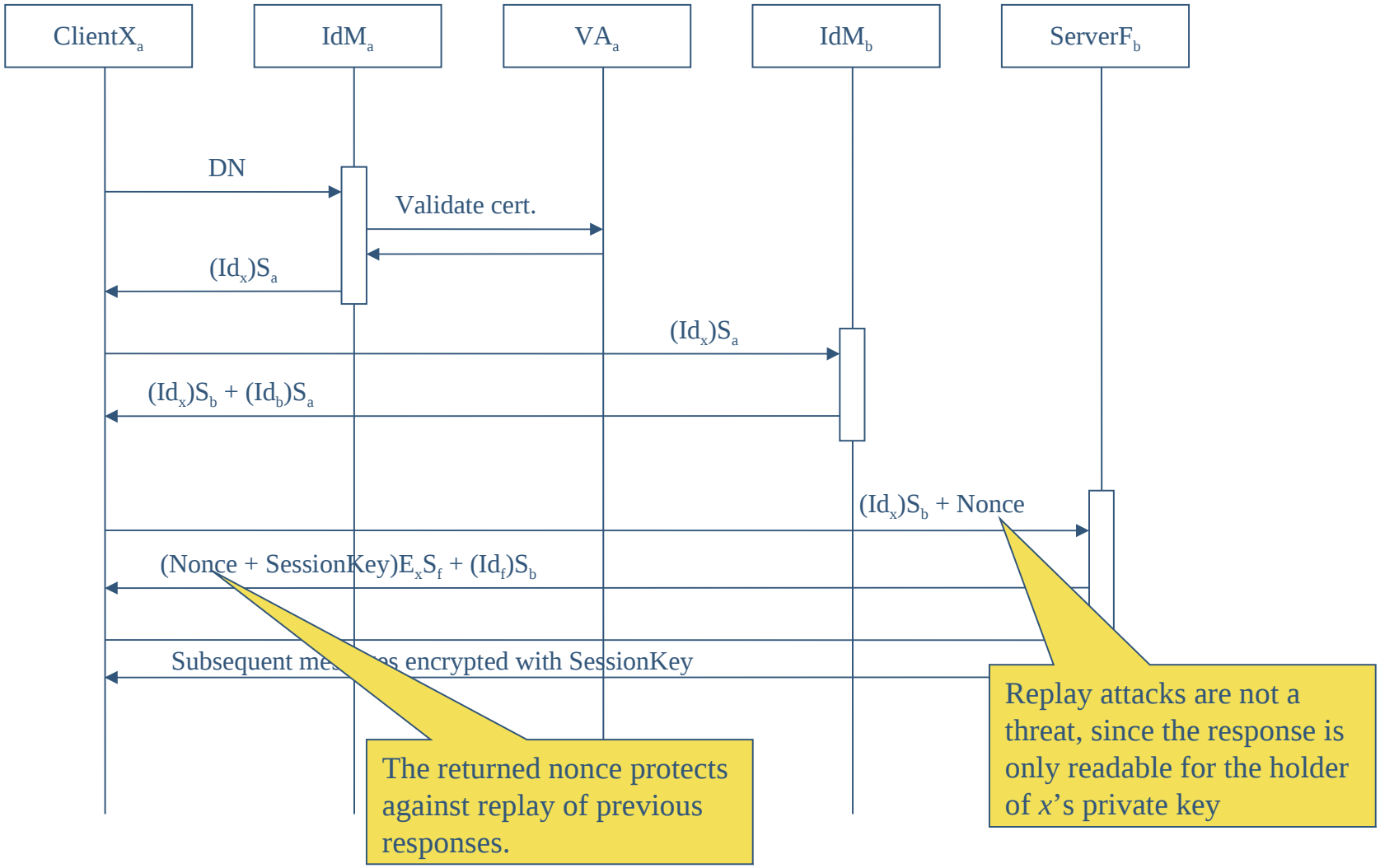
Step 3: Authenticated access to foreign server (stateless)



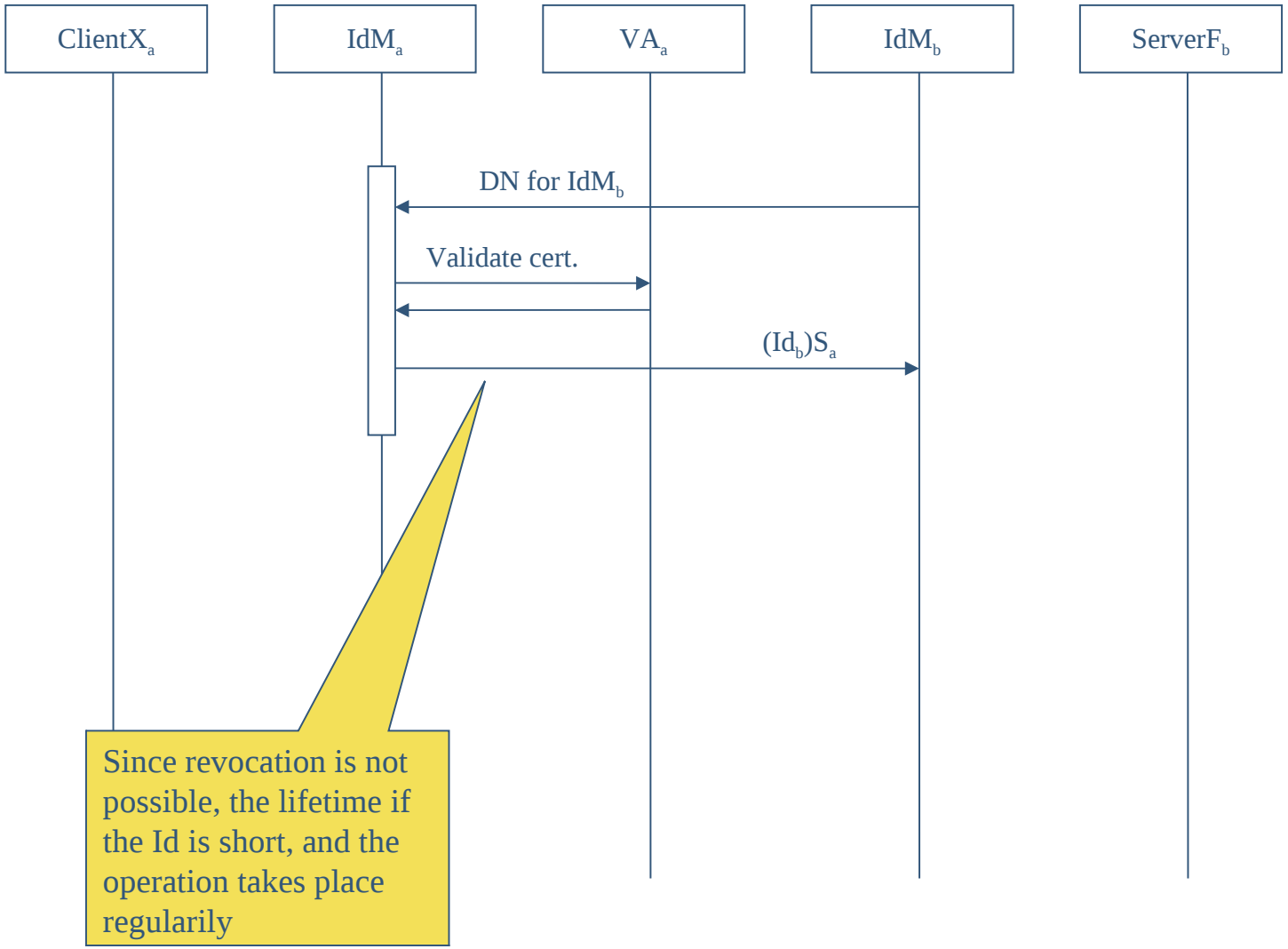
Step 4: Authenticated access to foreign server (stateful)



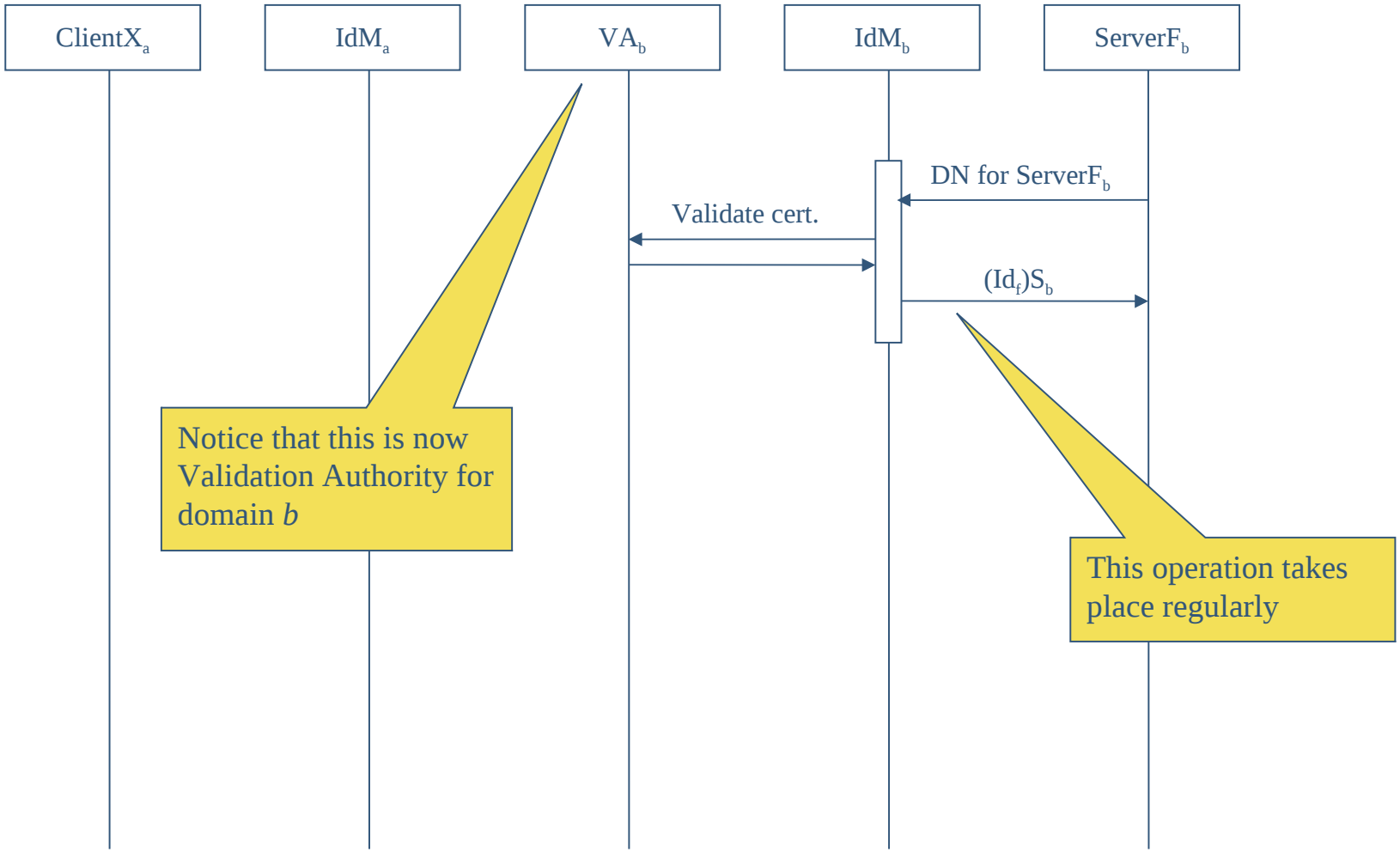
Step 5: Authenticated and encrypted access to foreign server (2-phase stateful)



Side Step 1: Issue a cross domain IdM Id



Side Step 2: Issue a server Id



Scalability issues

- Administrative tasks relate to local users and resources only
  - scales with the size of own domain
  - **no CRLs distributed across domains!**
- Interdomain trust is expressed as cross certificates.
  - number of certificates are approx. domains^2
- Complexity of policies must be managed
  - roles harmonized across domain borders

GISMO IdM for mobile units

- Based on the existing GISMO IdM

- Extended to include support for Android units
  - With poor support for SOAP-derived standard like WSSec, SAML etc.
- Employs a dual stack presentation layer
  - SOAP and serialized Java objects (POJO)
- Interoperability issued are addressed
  - Proxy nodes breaks SOAP/POJO separation

The first solution was

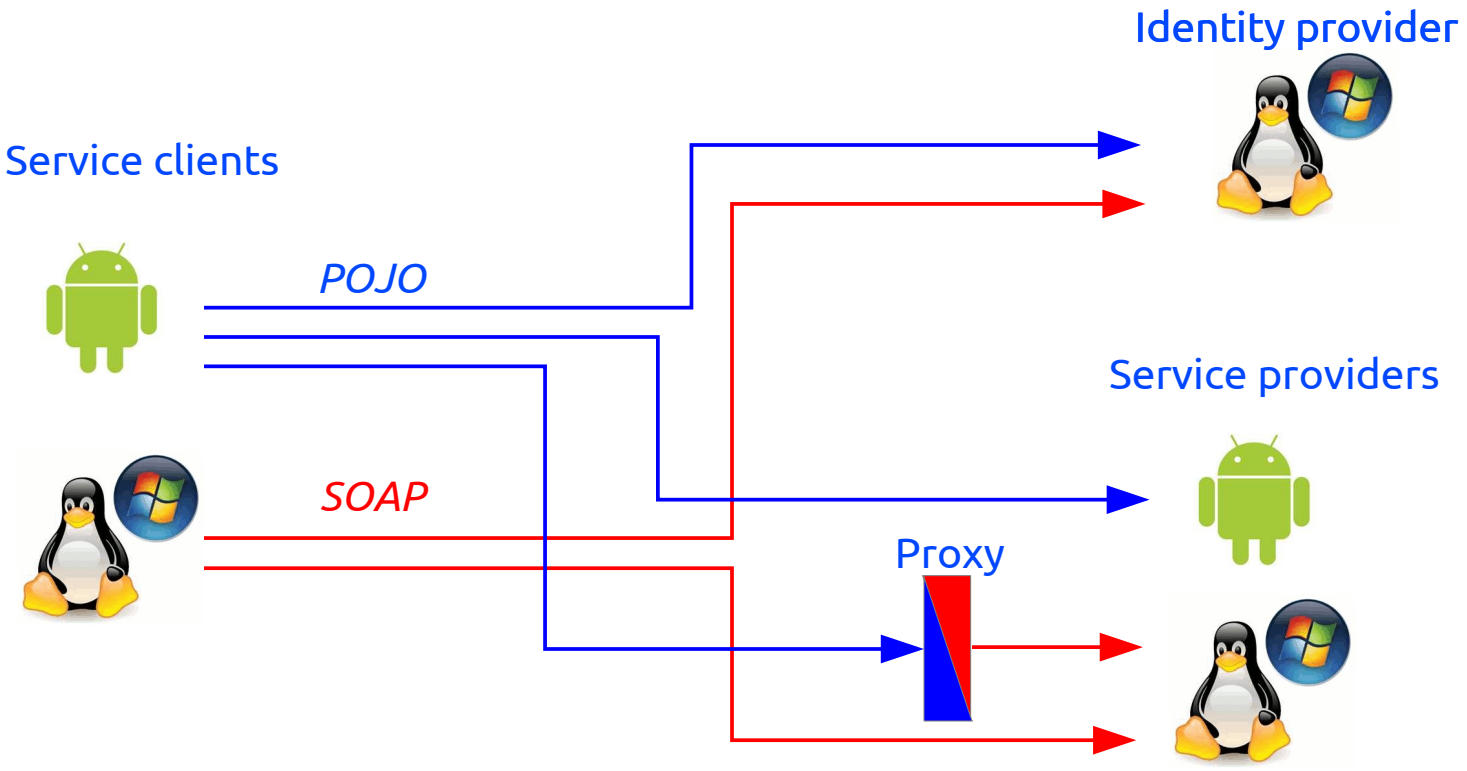
- Based on XML/WS standards
  - SAML, XACML, WS-Security
- Programmed in Java
  - Java-WS, Metro library
- Made as isolated/self-contained components

- *Not* portable to systems without extensive SOAP support.

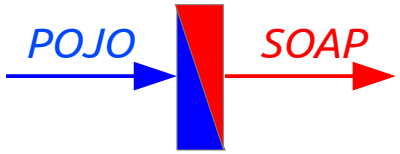
GISMO IdM for Android

- Includes clients and service providers based on the Android OS
  - Android has poor SOAP support
    - No Sun XWSS library (WSsec and SAML)
    - No Web Services architecture
- Employs a dual presentation layer (protocol syntax)
  - SOAP/WSsec/SAML
  - Serialized Java objects (POJO)
- Offers a lightweight service container
  - Well suited for service provisioning on Android
  - Employs the *class property* of a Java object

Invocation diagram



Conversion proxy



Conforms to the service component definition

Parameter class selects the service

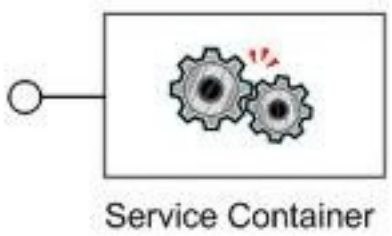
```
public class MainClass {
  public Serializable service(WeatherRequest wr,
                              Properties props) {
    try {
      Weather w = new Weather();
      String result = w.getWeatherSoap()
          .getWeather(wr.town);
      return result;
    } catch (Exception e) { return e; }
  }
}
```

Service class generated from WSDL-compilation

NB: A conversion proxy breaks the signature.

Service Components for POJO

During run time
(introspection)



init

service*

Jar files
Manifest.mf: Main-Class...
Required methods:
p3 service(p1,p2)
void init()

Stored in a given directory

The class of the parameter is associated with the service class. The client does not name the service, but simply supplies the service call with the correct parameter class.

The light weight framework for service provisioning allows for Android units to offer services

CLIENT CODE:

```
try {
   ClientProxy client = new ClientProxy(prop);
   WeatherRequest p = new WeatherRequest(); // parameter
   p.town = "CHICAGO";
   Object s = client.invokeApplicationService(p);
   return s.toString();
} catch (Exception e) {...}
```

Support for Access control

- Attested subject attributes can be used for
  - Roles in an RBAC context
  - Attributes in an ABAC context
  - Preferences for applications
- Maintenance of subject attributes are done in the IdM, not PKI
  - they reflect the policy of the COI, not PKI domain
- Cross-COI operation requires harmonization of attributes
  - names and values must mean the same everywhere
  - not necessary to exchange all attributes, only those significant for the cross-COI operation

Interoperability issues

- GISMO IdM only speaks «to itself»
- Turns into a *portability* issue
  - How can the protocols be implemented on other systems?
  - Present version written in Java
    - Ports well to Windows, Linux etc.
      - To the extent allowed by the libraries
  - Port to Python, .NET etc.
    - In «principle possible», but yet to be done
    - Relies on the availability of SOAP libraries (unknown)

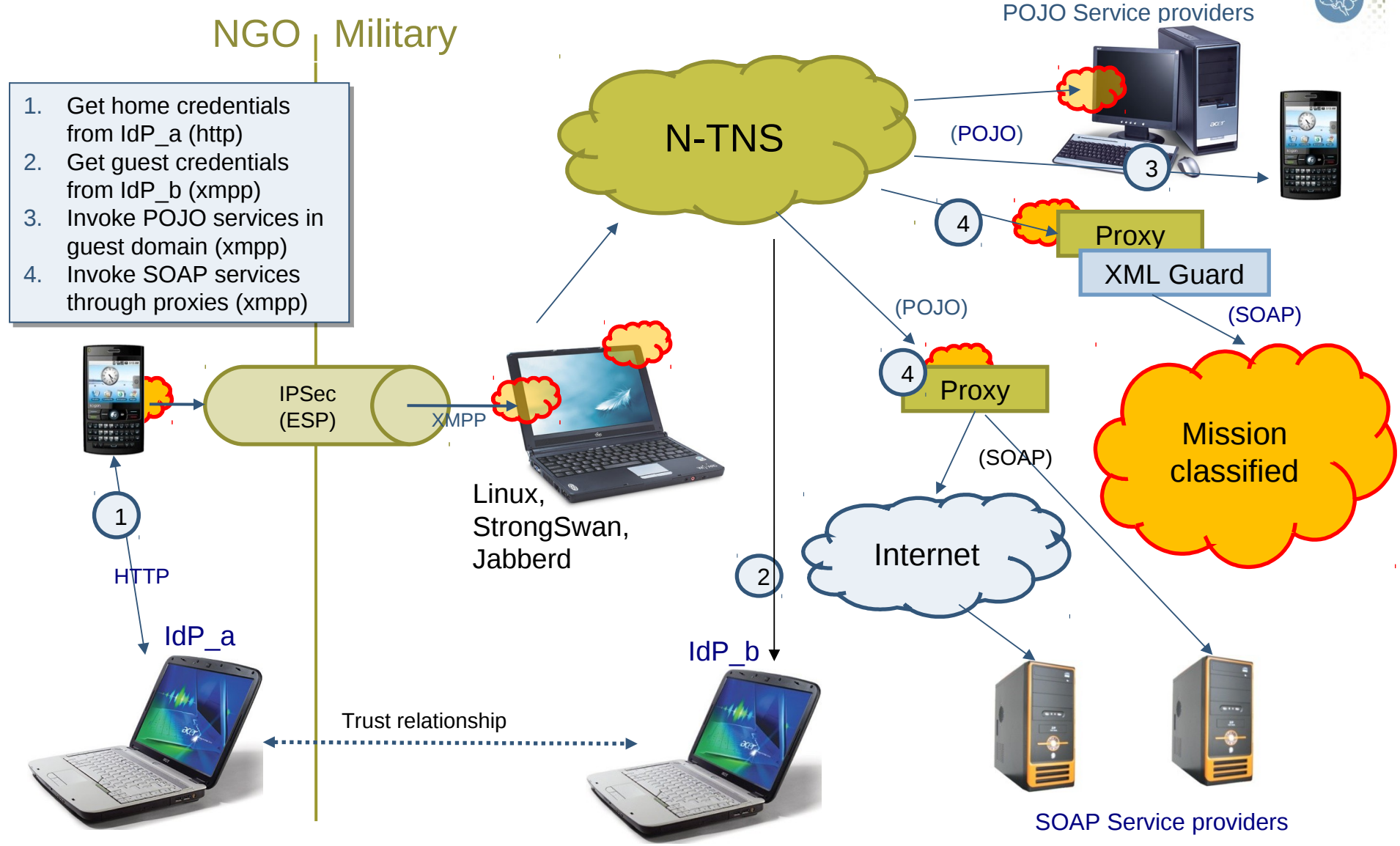
- *Claim: POJO- and SOAP-based protocols are equally portable*

Experimental evaluation

- Gismo IdM has been evaluated for correctness and feasibility
- A large field experiment will take place in June 2012
  - Opportunity to test stability, performance and scalability

= Colored enclave

Protected service invocations for Android



Principles of PubSub

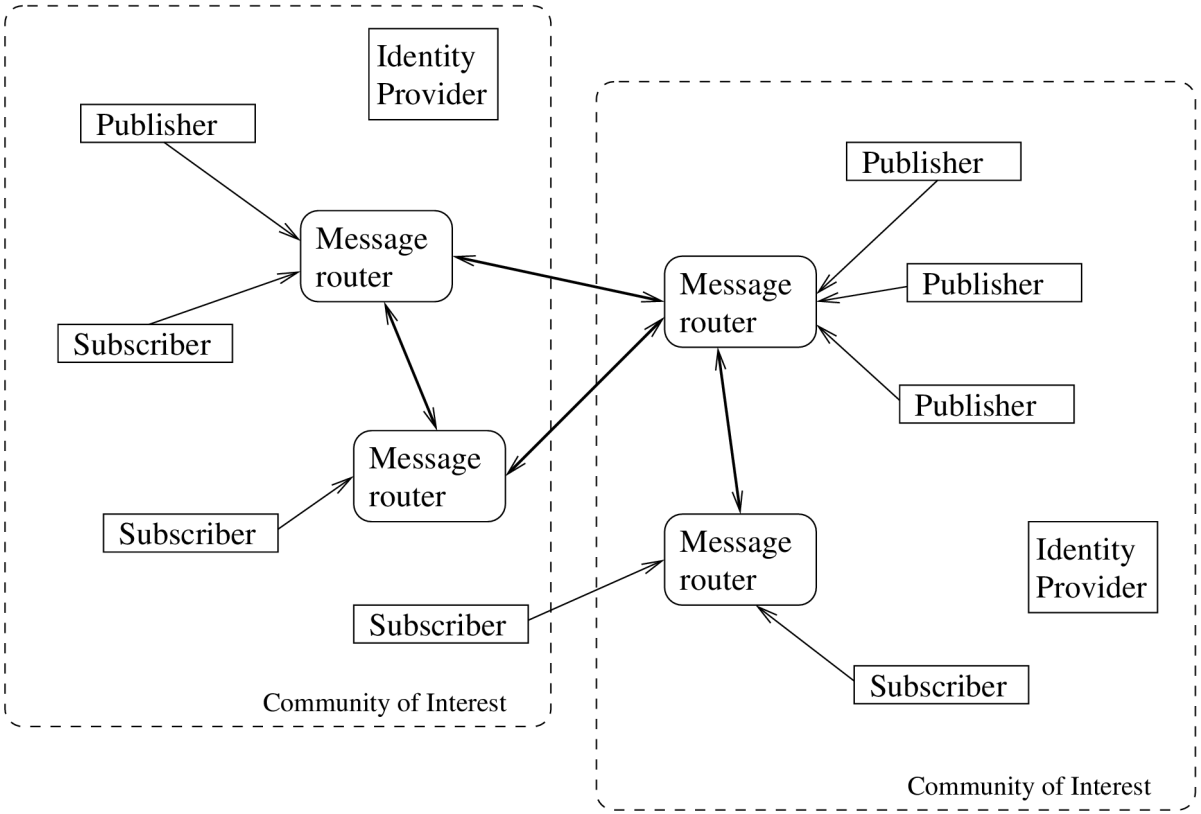
- Messages are not individually addressed
- Routed from producer to receivers based on content
  - aka content routing
  - content metadata (topics)
- Publications are annotated with *topics*
- Receivers *subscribe* to topics

- Message transport is *asynchronous*
- Scales better
  - employs multicast topologies
- Smaller resource consumption
  - Messages are queued outside the hosts

PubSub security problems

- Is the published information authentic and unmodified?
- Who is the publisher, is it authorized?
- Will the published information only reach authorized subscribers?
- Who operates the message routers? Are they operating correctly?

Extending the PubSub network with Identity Management



Principles for PubSub security

- The confidentiality is the concern of the *Publisher*
  - should be able to express *subscriber requirements*
- The integrity is the concern of the *Subscriber*
  - should be able to express *publisher requirements*

- The authenticity of identities is the responsibility of the *Message Routers*
  - through identification of clients and peer message routers
- The correctness of keys and credentials is the responsibility of the *Authority*
  - establishes identities, issues and revokes keys and credentials

How to express a subscription?

- One or more topics
  - must match (hierarchically) the topics of the publication
- A publication requirement
  - a boolean expression evaluated with respect to the publisher's identity attributes
    - must evaluate to *true*

Subscription

topics:
/news/security/incidents
/alarms/ids/building−203

publisher requirements:
(nationality="uk") or
(sec_clearance="NATO secret")

IdM and Integrity Control

- Trust in security operations relies on service integrity
  - unadulterated software and hardware
- My means of hardware, a sterile configuration can be *sealed*
  - checked at botstrap or anytime

$$hmac = f\left(K, h\left(mem\right), challenge\right)$$

  - attested by an IdP and validated by anyone
- Can relax other sequrity requirements

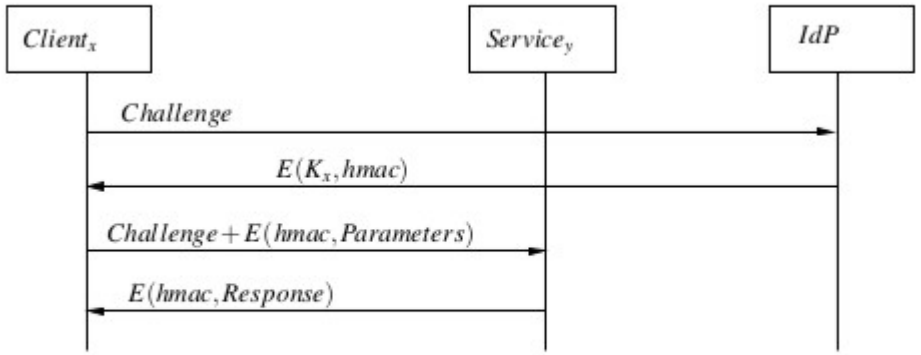


Figure 1. The protocol for protection of genuineness, based on symmetric crypto keys and the HMAC function given in Equation 1.

Publication from these efforts

- "*Identity Management Without Revocation*", Securware 2010, Venice, Italy, July 2010
- "*Architecture Patterns for a Ubiquitious Identity Management System*", under review for ICONS 2011, St. Maartens, January 2011.
- *"Identity Management for Android"*, Securware 2011
- *"Federated Identity Management in a Tactical Multi-Domain Network",* Int. Journal on Advances in Systems and Measurement, 2011 vol 4 nr 3&4
- "Identity Management and Integrity Protection in Publish-Subscribe systems", under review for MILCOM 2012
- "Identity Management and Integrity Protection in the Internet of Things", under review for EST 2012
- "Protected and Controlled Communication Between Military and Civilian Networks", under review for MCC 2012

Summary and Conclusions

- An IdM should support authentication as well as access control
- An IdM should not revoke “live” identity statements
- An IdM MUST support mutual authentication
- An IdM should provide loose coupling between domains and maintain autonomy
- Android has poor support for SOAP, SAML and WSSec.
- A separate presentation layer can be defended
  - Serialized Java objects have been chosen
  - Flexible, efficient and intuitive API
- Invocation “across” presentation layers (POJO-SOAP) is provided by proxy agents
- Future research includes an international field experiment