Christophe Andrey

Certificate-based Authentication for CORBA

Diploma Thesis

LABORATOIRE DE SYSTEMES D'EXPLOITATION

Goals

- Implement Authentication in JacORB with SPKI certificates.
- Certificates carry a subject‘s authenticated attributes.
- Secundary goal: Review existing certificate infrastructures.

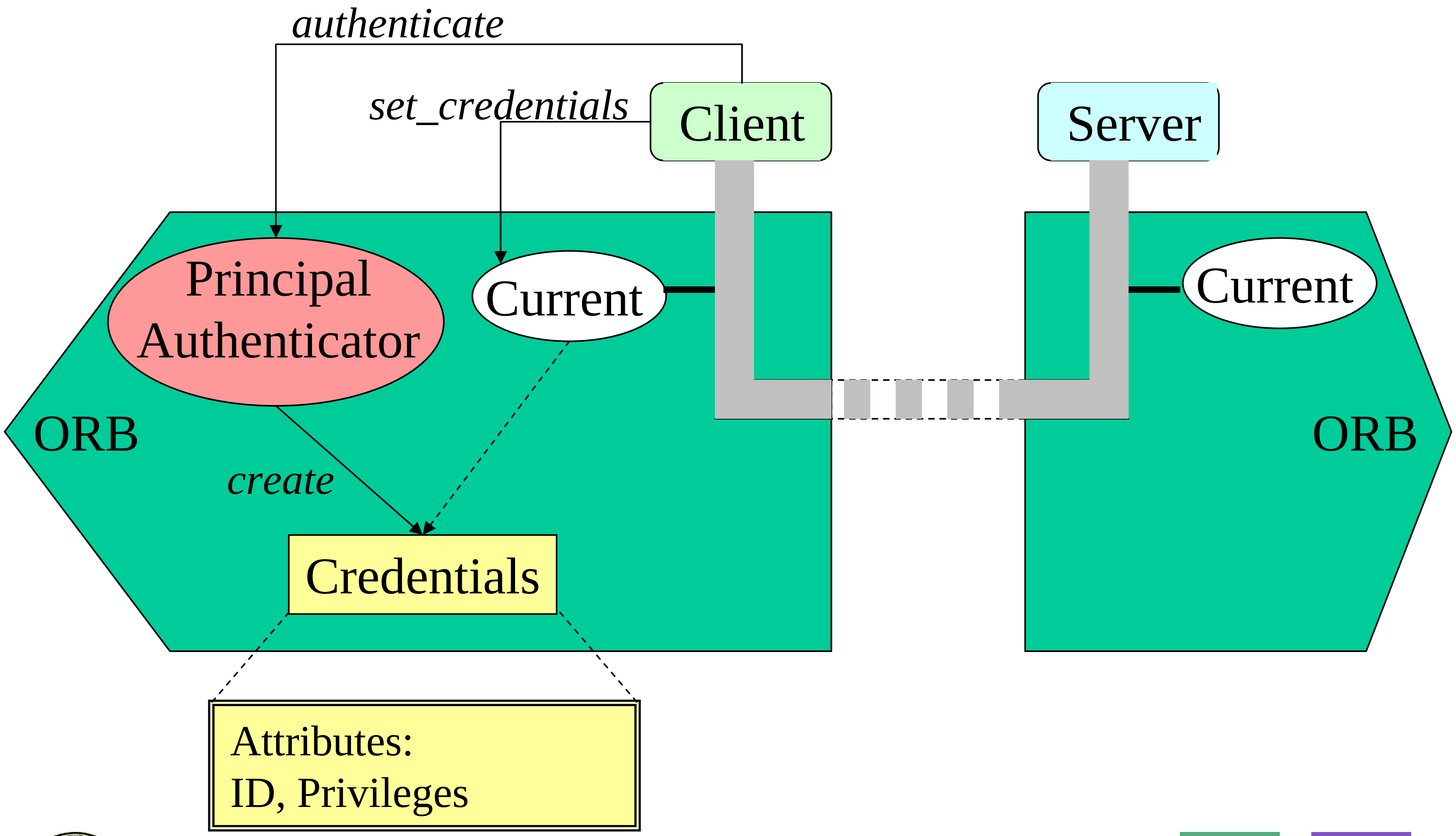
Structure

1. Theoretical background
   - JacORB
   - Authentication in CORBA: Credentials
   - SPKI
2. Protocol:
   - Creation, retrieval and transmission of authenticated credentials
3. Implementation: layered architecture
4. Demonstration: *Access Control* for the name server
5. Contribution

JacORB

- Free and pure-Java implementation of CORBA
- By Gerald Brose, FU-Berlin
- No security service (Version 0.9)

Authentication in CORBA



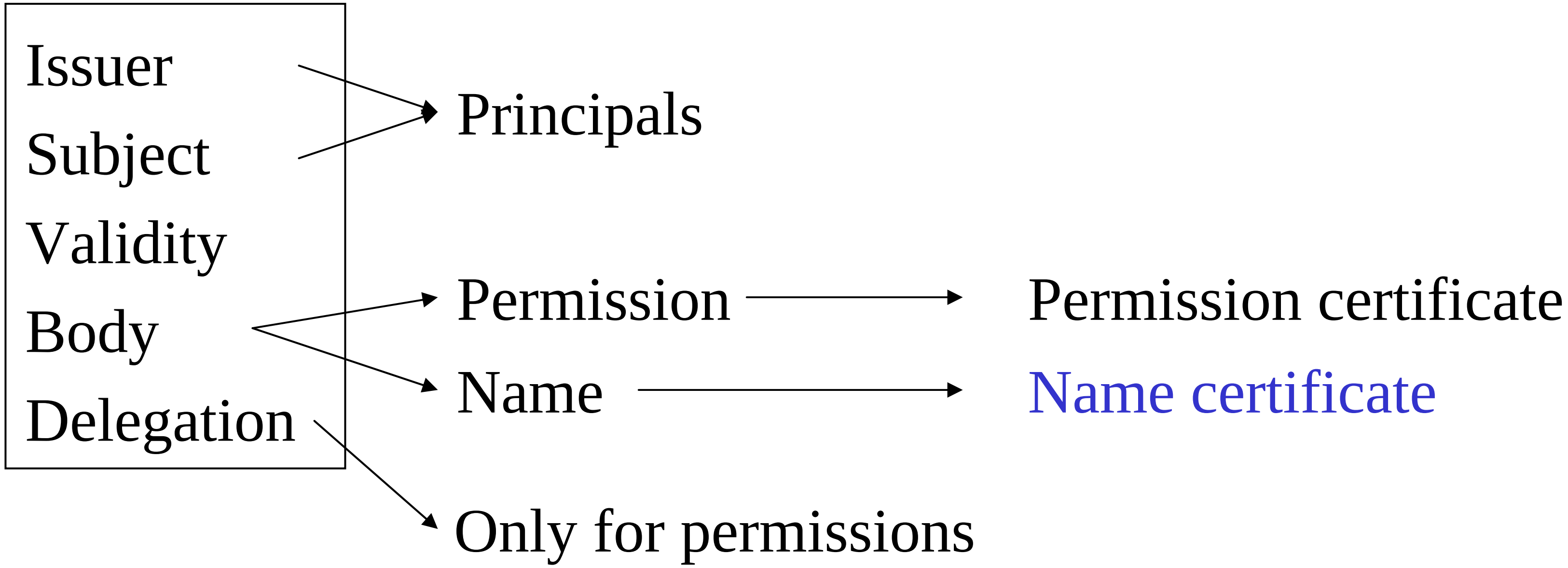
Credentials

- Not specified by CORBA

- Meaning / content → ID attribute
- Format → SPKI certificate
- Origin → protocol

SPKI‘s structure

- Infrastructure:
  - X.509 & PGP : 1 principal = 1 name
  - SPKI : 1 principal = 1 public key
- Structure of a certificate:



Reasons for choosing SPKI

- Novelty
- More expression power than X.509 or PGP
- Flexible trust model

SPKI Grammar

- Based on S-expression
- 83 production rules
- Unstable
- Example
- Advantage: Interoperability
- Disadvantages: associated with parsing
  - Large *Lookahead* (up to 14)
  - Two Forms: advanced and canonical

Example of an S-Expression

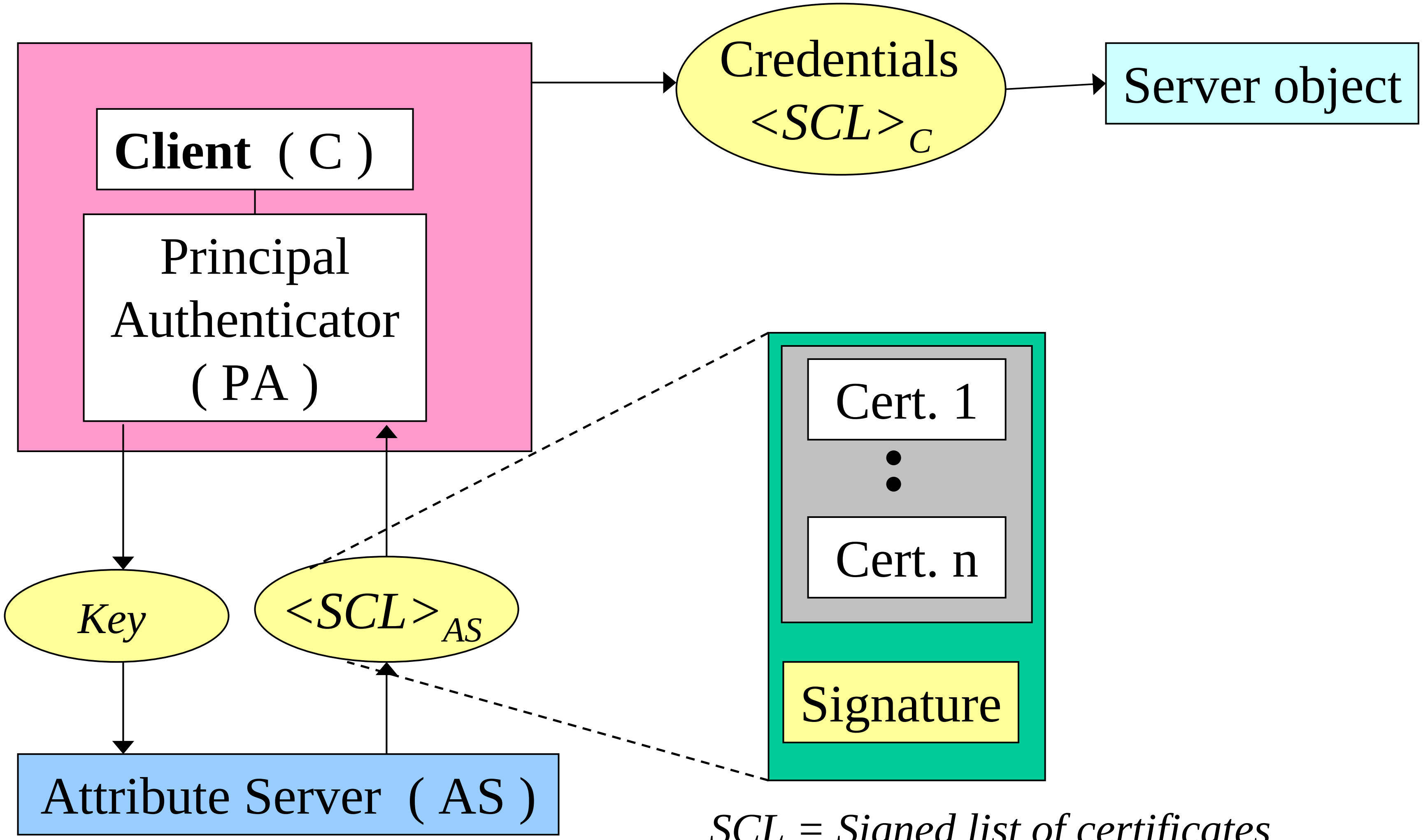
```
(sequence
    (cert
       (issuer
          (name
             (public-key rsa-pkcs1-md5
                    (n  ALH467KORQkeigyGhMRAwYfHxWyfLmO++tC3WJaasUp7becE0H7aWXay9jlunB8M
                        JixayaAxZKXmZ/pU17UuwMpLlxAeY3BAq2Mdjhcdwgqt25+CwGYOH0xyL8dGTePn
                        14OH4+cj5/rDNA/y2zWF6T6isXPHneEi1U23EU1WgeR7 )
                    (e AQAB ))
             AccessId/1 ))
       (subject
          (public-key rsa-pkcs1-md5
                    (n AI8RDzo1NkvlhvmGcQtUC6VPgVXFaYdap1pDZtfnHqE4avTPtRiw1QXqDrlpRQsp
                       M+h3xfZ7yFAxlK5MOFcRGlcdykhqbr7lshyyHcme3+9reJYhz7taik9OUDLjzNeg
                       WCkEPnhk2GrgT5h1JUz25yh97c7fyjiWraF8W2hDy0Vd )
                    (e AQAB )))
       (not-before 1999-03-08_11:52:31 )
       (not-after 1999-03-08_12:22:31 ))

    (signature
          (hash md5 Dt3V2QCqn0WT7/mfN0hAhA== )
          (public-key rsa-pkcs1-md5
              (n ALH467KORQkeigyGhMRAwYfHxWyfLmO++tC3WJaasUp7becE0H7aWXay9jlunB8M
                 JixayaAxZKXmZ/pU17UuwMpLlxAeY3BAq2Mdjhcdwgqt25+CwGYOH0xyL8dGTePn
                 14OH4+cj5/rDNA/y2zWF6T6isXPHneEi1U23EU1WgeR7 )
              (e AQAB ))
          XtIoC+RMtouXCv69Kq/tOcUTUqMDq+cf5wd1urkBQoZuvhwSVcHE6gv9wqY8FnCn
          o0Cyu+ZSY1PLVwUMQjvdZEwHieDRDWTeiyDinVUGwUKo0mlP9d9rJjUCnKh37P8J
          92oslUVy8kxjXtNZsIap3nOc9RTvKoh69gDcrW7QcuQ= )
)
```

Struktur

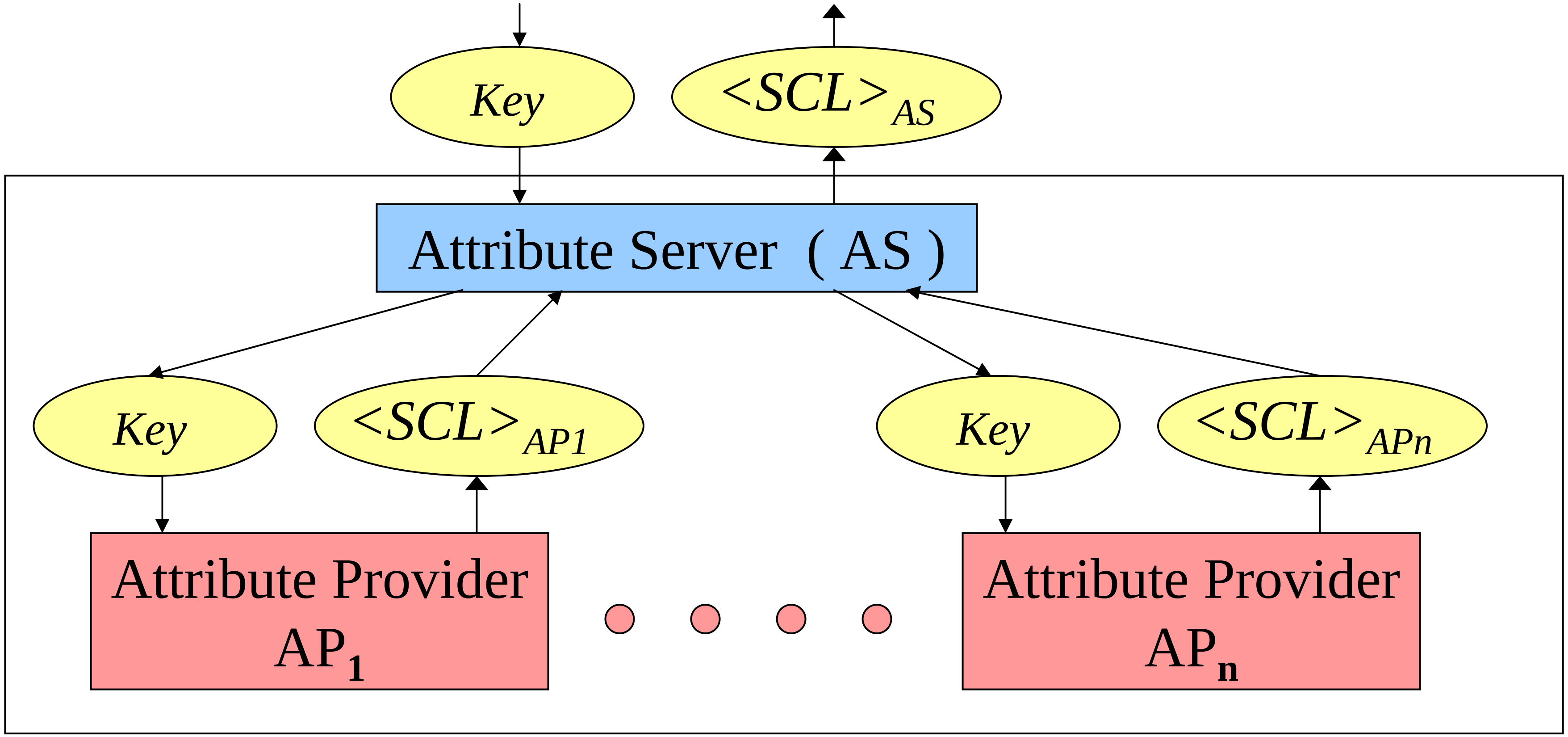
1. Theoretical background
   - JacORB
   - Authentication in CORBA: Credentials
   - SPKI
2. Protocol:
   - Creation, retrieval and transmission of authenticated credentials
3. Implementation: layered architecture
4. Demonstration: *Access Control* for the name server
5. Contribution

Protocol: Overview

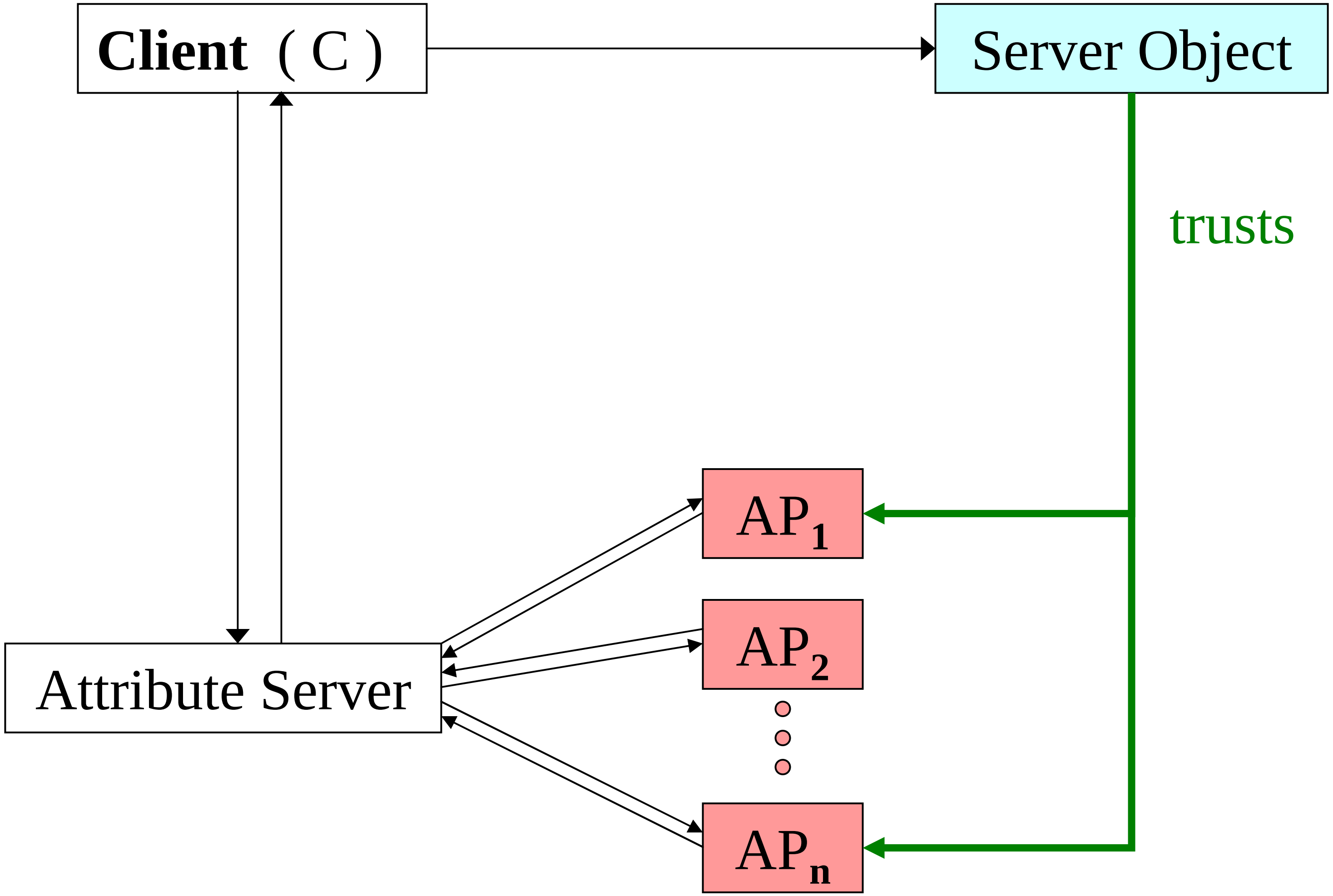


SCL = Signed list of certificates

Protocol: Structure of the Attribute Server



Protocol: Trust Relations



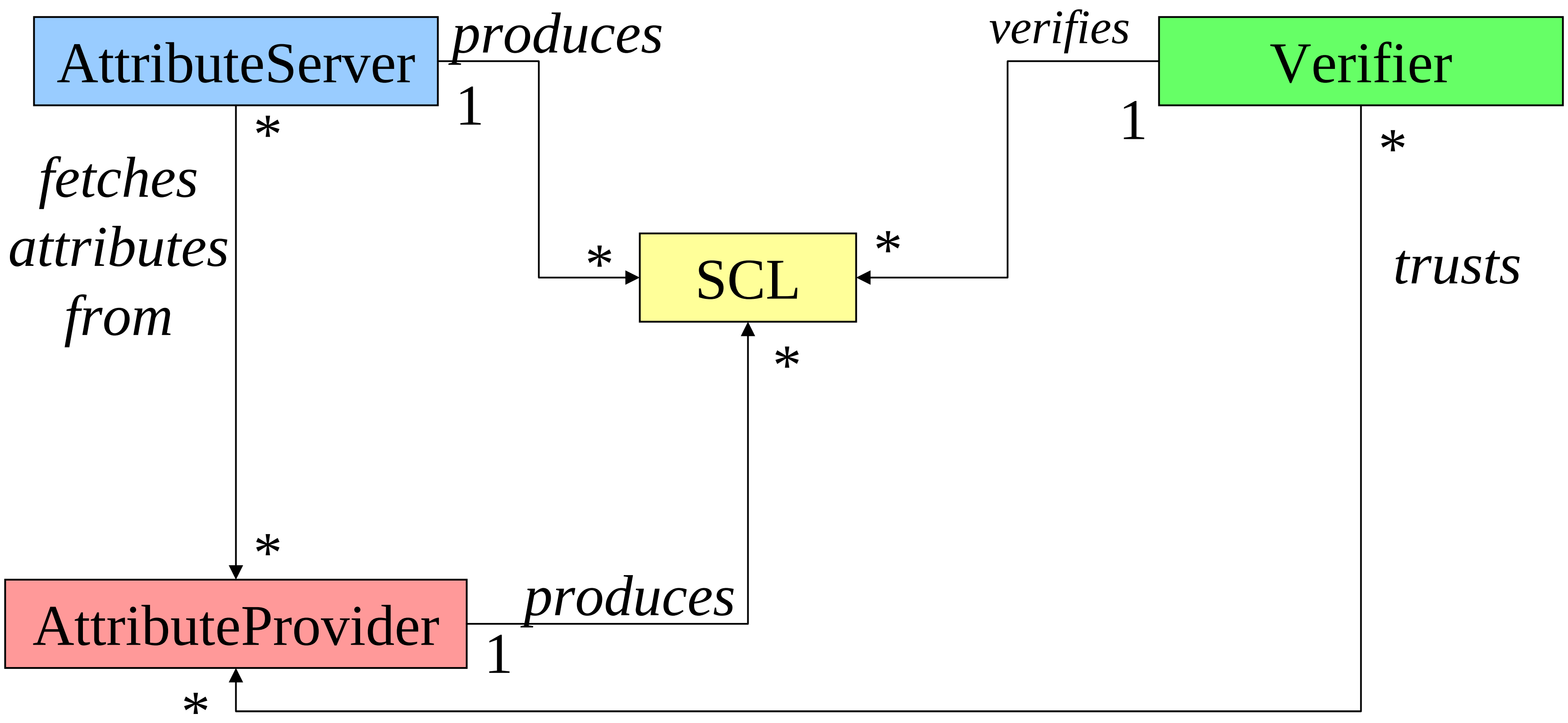
Struktur

1. Theoretical background
   - JacORB
   - Authentication in CORBA: Credentials
   - SPKI
2. Protocol:
   - Creation, retrieval and transmission of authenticated credentials
3. Implementation: layered architecture
4. Demonstration: *Access Control* for the name server
5. Contribution

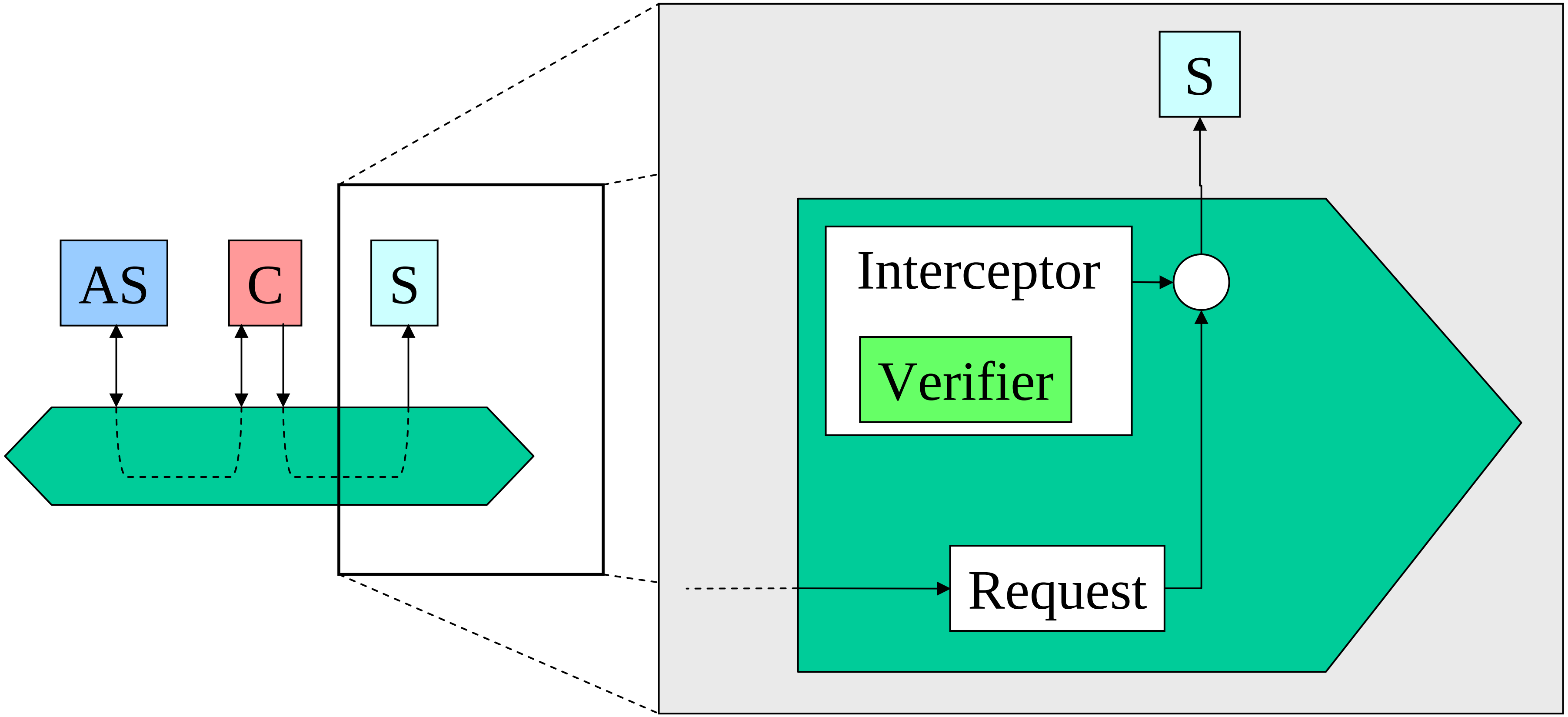
Architecture of the Implementation

5	**JacORB**	`jacorb.*`
	Principal Authenticator, Current, Credentials	
4	**Authentication**	`spki.auth`
	Attribute server and attribute provider	
	Creation and transmission of the credentials	
3	**SPKI certificates**	`spki.certificate`
	Signature generation and verification	
	S-expression ↔ certificate	
	Generation of key pairs	
2	**S-expression objects**	`spki.sexp`
	Serialization in canonical / advanced Form	
	Reading ASTs	
1	**Parsing**	`spki.parsing`
	Syntax trees	`spki.syntaxtree`
	Visitors	`spki.visitor`

Layer 4: Object Model



Layer 5:
Mechanism on Server Side



Layer 3: Functionality

1. Notation-independent representation of certificates

2. Signature and hash engines

3. Generation of keys for principals

4. Integration with Java 2 API

Layer 3: Main Classes

- Certificates
  - 2 categories: name and permission certificates
  - No integration with Java 2
  - Each certificate is associated with two principals: its subject and its issuer
- Principals:
  - Name
  - Hash value
  - Public key
- Public key
  - 3 types : RSA-SHA1, RSA-MD5, DSA-SHA1

Layer 3: Keys‘ Functionality

		Java	Cryptix	my impl.
Representation	public keys	●		●
	private DSA	●		●
	private RSA	●	●	
Signature engine	DSA	●		
	RSA		●	
Key pair generation	DSA	●		
	RSA		●	

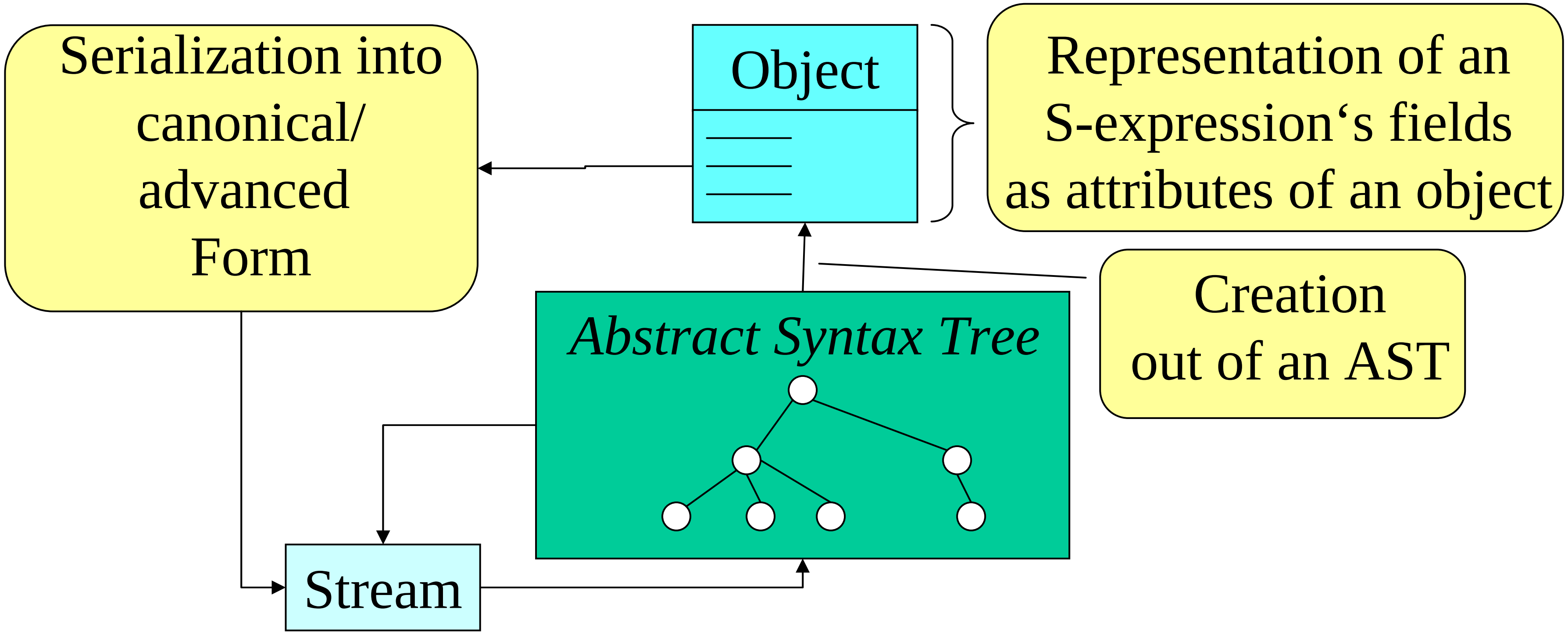
- Persistence:
  - File containing the S-expression of the key pair
  - Private key is encrypted with a pass-phrase

Layer 1

- Functionality: Parsing S-expressions
- Parser generator: JavaCC and Java Tree Builder (JTB)
  - has generated all classes in layer 1
- 3 packages: spki.parsing, spki.syntaxtree, spki.visitor
- Visitors:
  - Advantage: Implemented functionality remains despite modifications of the grammar.
  - Disadvantage: Not adapted to local operations
- Problem: syntax trees are burdensome to explore
- Solution: An intermediate layer that represents structured S-expressions as objects.

Layer 2: S-Expression Objects

- Functionality:



Serialization into canonical/ advanced Form

Object

Representation of an S-expression‘s fields as attributes of an object

Abstract Syntax Tree

Creation out of an AST

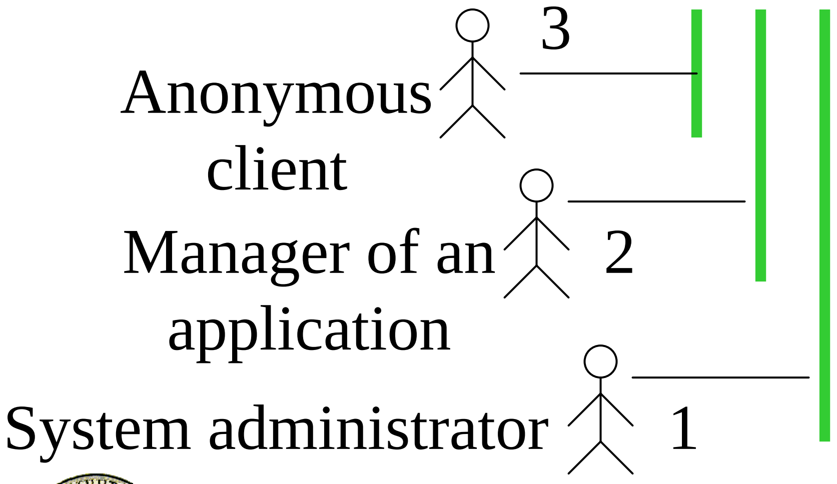
Stream

- A subset of all S-Expressions is supported.

Example Application: Principle

- Goal: Access control of JacORB‘s name server, based on authentication.
- Rationale: The name server is security-critical.
- Each client has 1 attribute: an ID

- Example:



3

Anonymous client

Manager of an application

2

System administrator

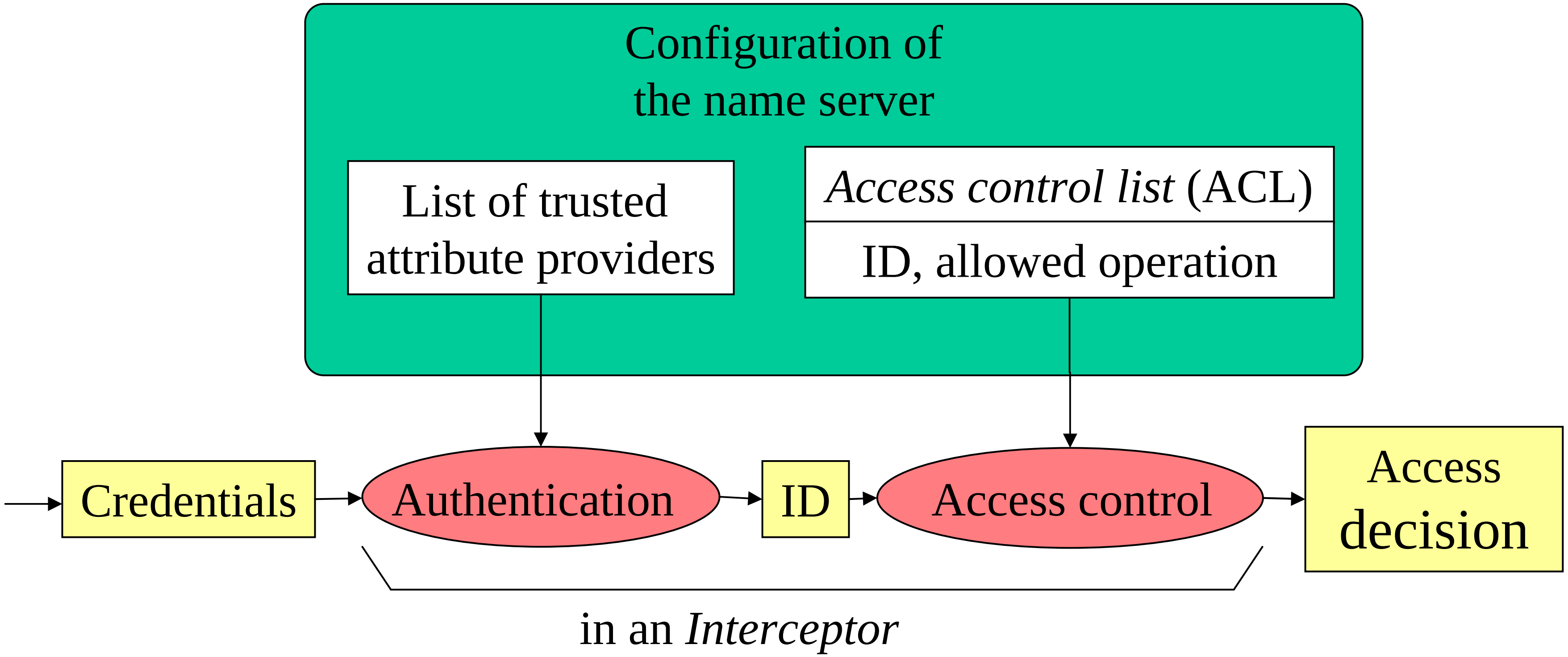
1

Operations on the name server

resolve (name) : object
list ( )
bind (name, object)
rebind (name, object)
unbind (name)
destroy()

Example Application: Mechanism

- Process on server side:



Contribution

- Knowledge
  - Real-world application of SPKI certificates
  - Demonstration that they are adapted to a security-critical application like authentication in CORBA
- Deliverables
  - A Java library for the serialization of SPKI certificates
  - Authentication in JacORB