Security model for hybrid token-based networking models

By Rudy Borgstede

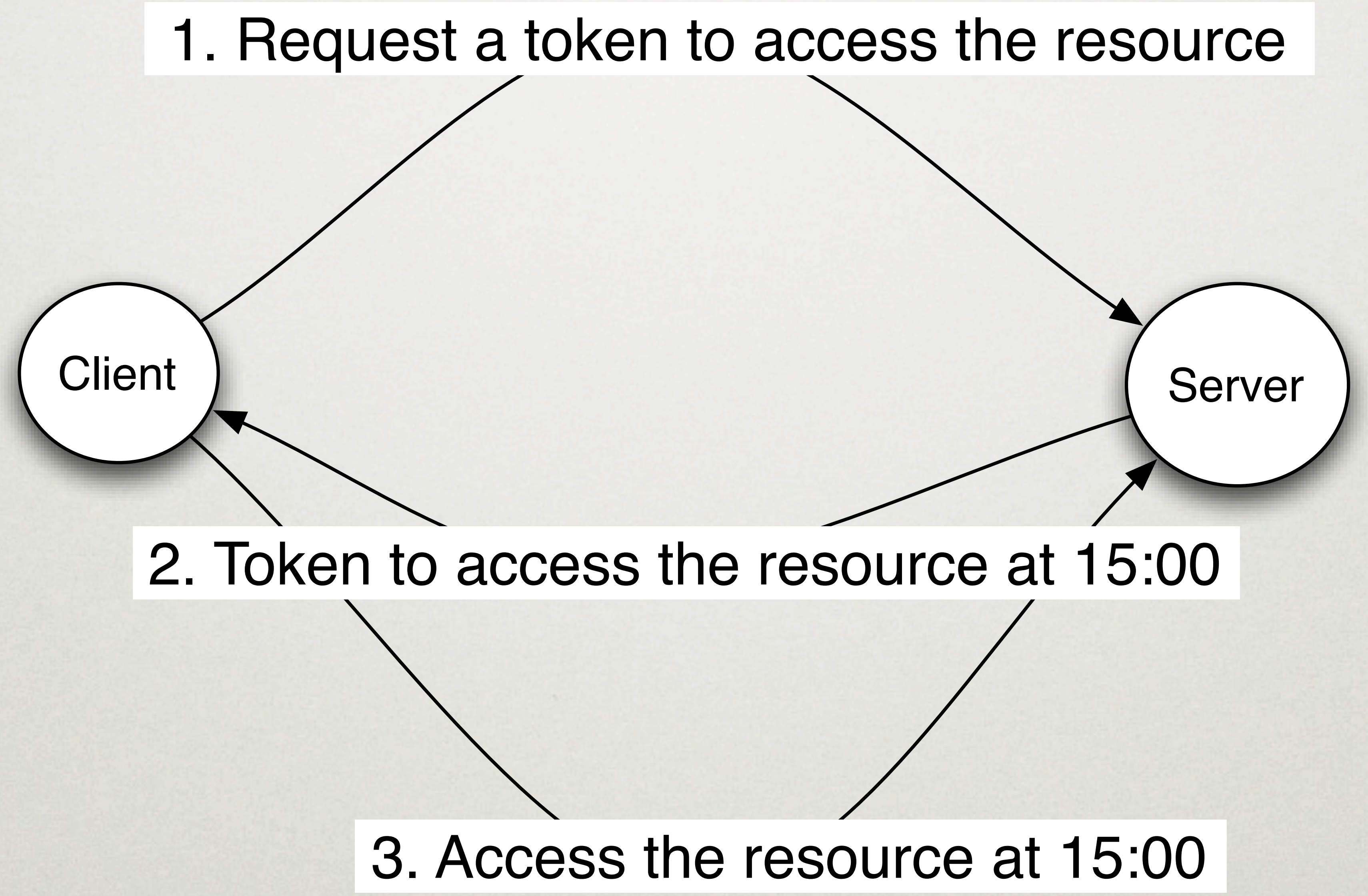
Contents

- Project Background
  - Complex Resource Provisioning and Token-Based Networking
  - Token Validation Service, the Java Aaauthreach project
- Identity-Based Cryptography
  - Public Key Cryptography and IBC
  - Public Key Infrastructure vs IBC
- IBC implementations and the Eyebee experiment
- Should we use IBC?

Complex Resource Provisioning

- Lookup Resources
- Composite the resources
- Resource Reservation
  - Global Reservation ID (GRI)
  - Policy
- Deploy

Token-Based Networking



1. Request a token to access the resource

Client

Server

2. Token to access the resource at 15:00

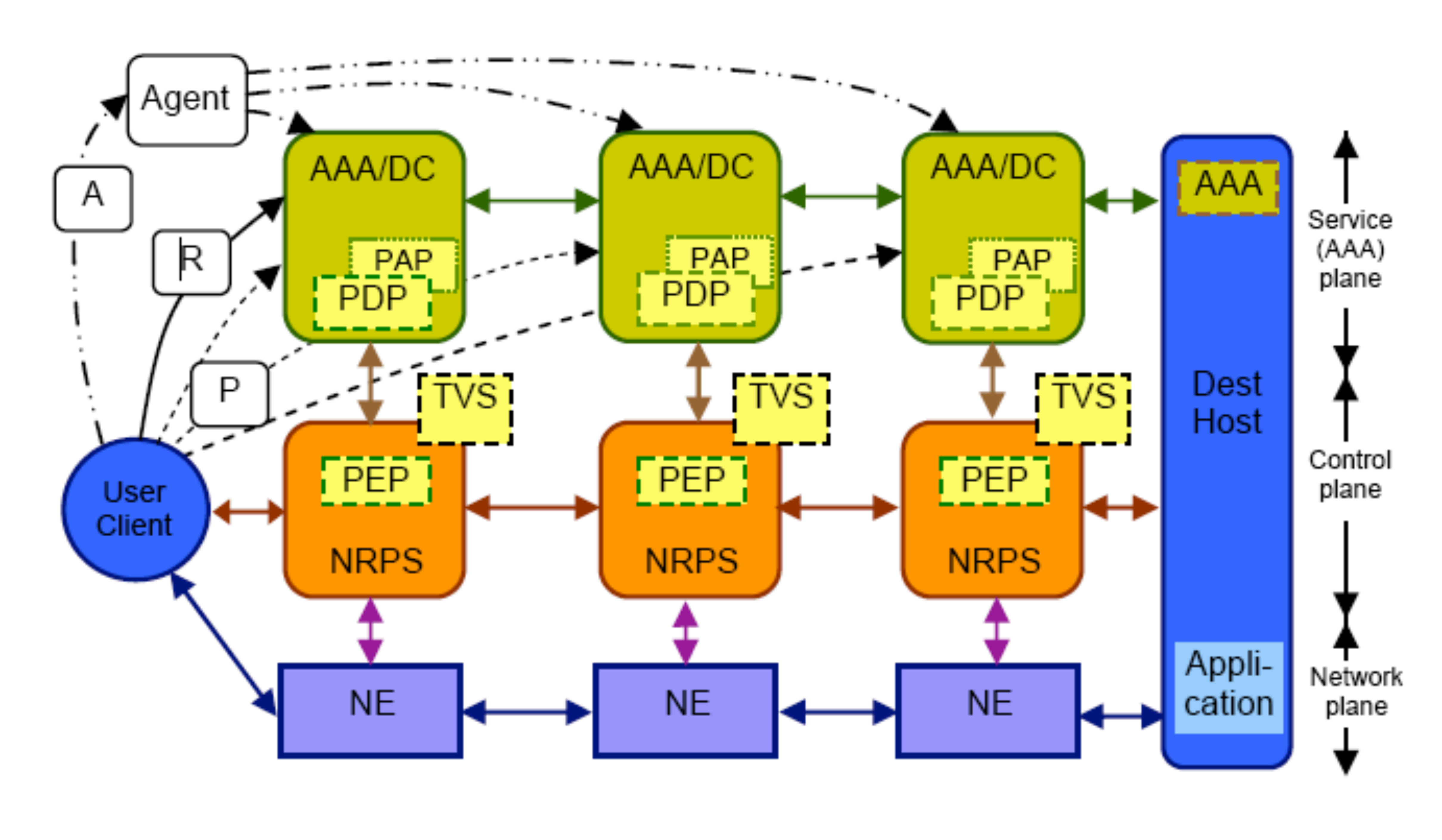
3. Access the resource at 15:00

TVS, the Java aaauthreach project

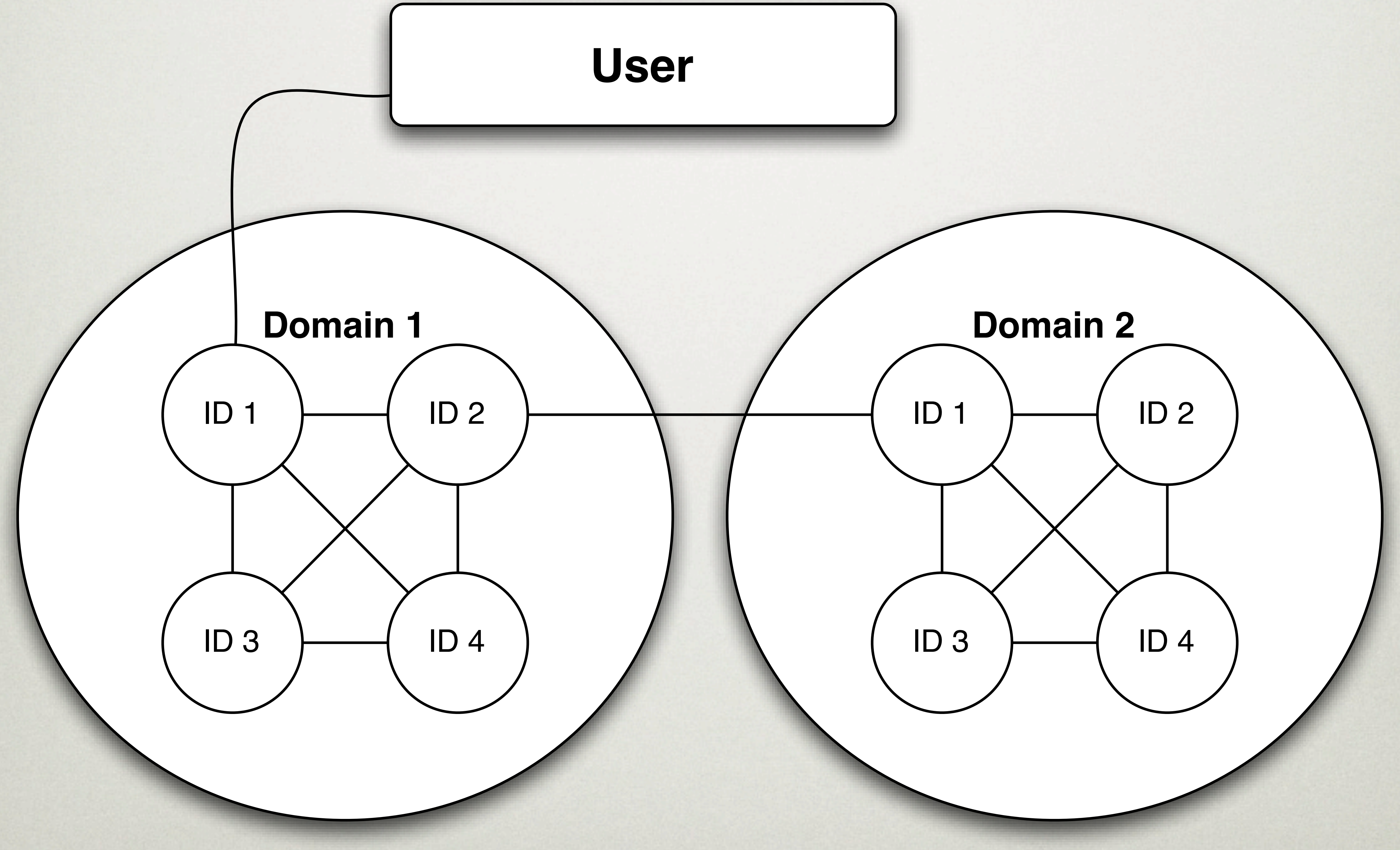
- TVS is a component of the TBN policy enforcement infrastructure
  - Manage resources
  - Manage reservations
  - Routes the tokens

- TVS is implemented as a pluggable component called the Java Aaauthreach project

CRP operational model

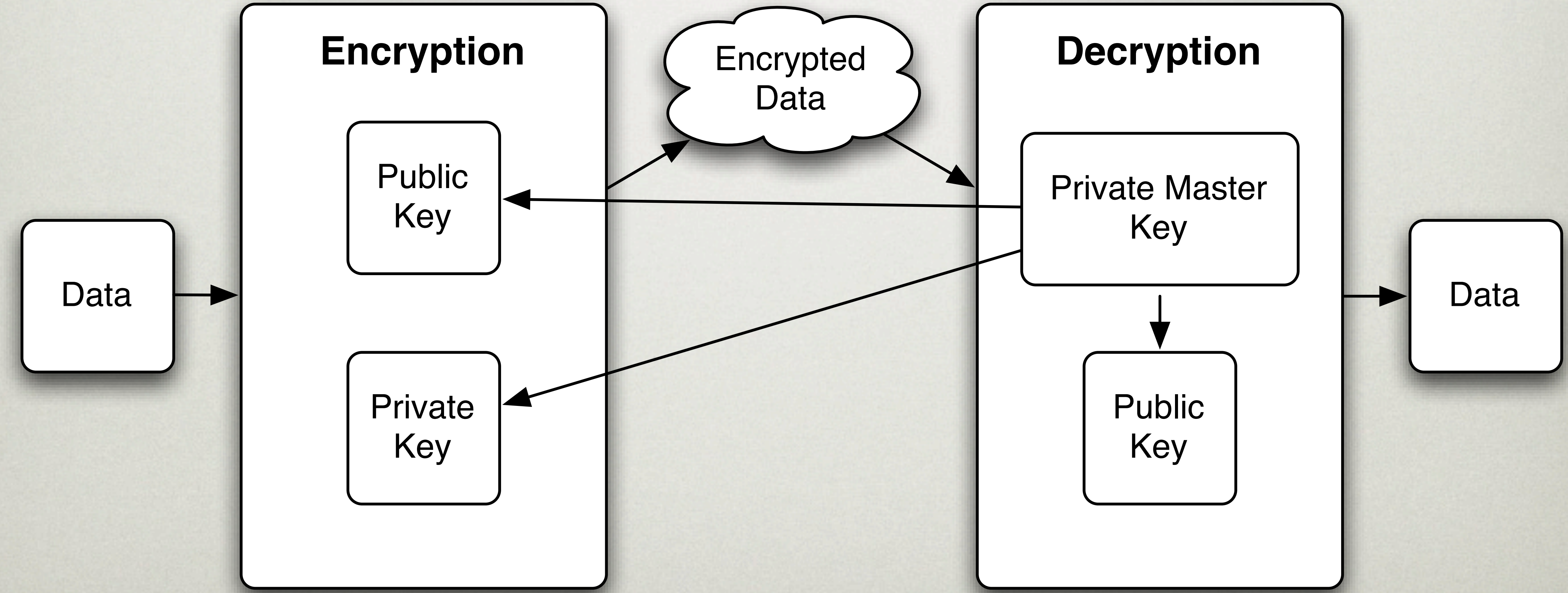


CRPS EXAMPLE



Public Key cryptography

- Private Master Key
- Private Key
- Public Key



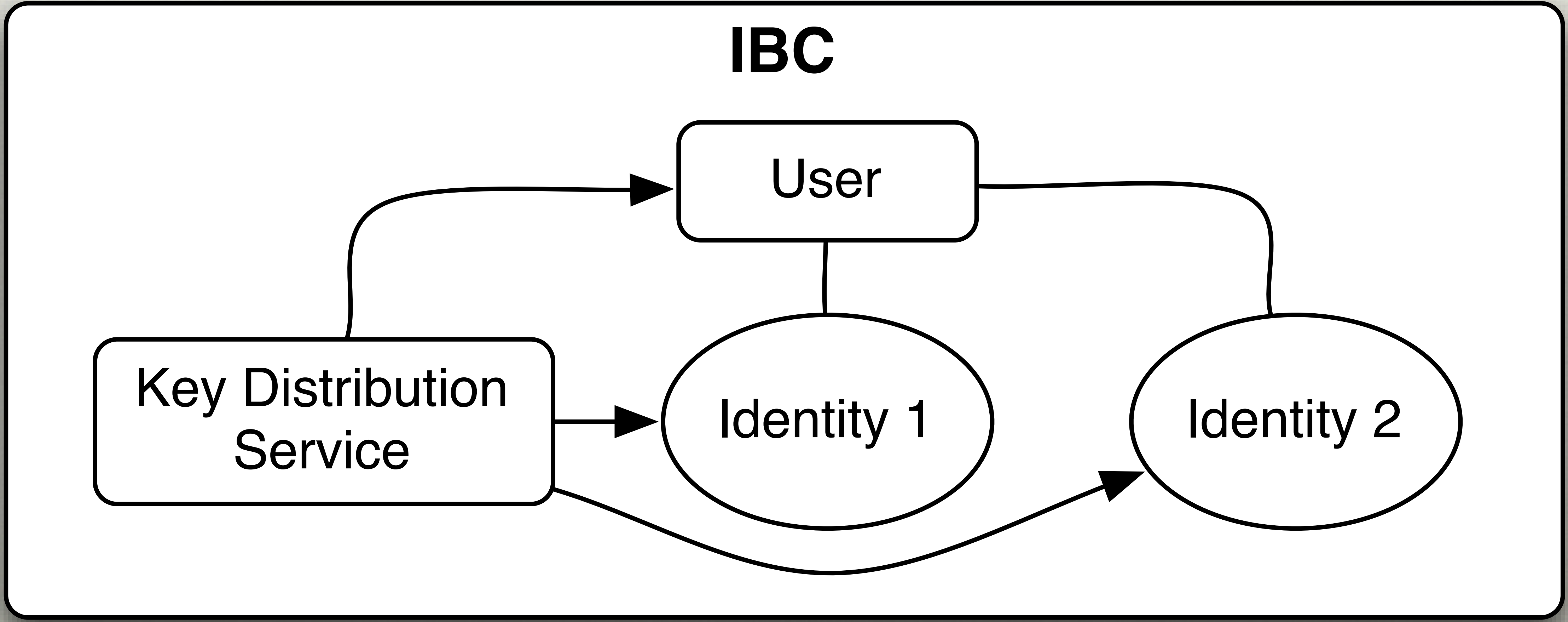
Identity-Based Cryptography

- Public Key is based on the identity of the destination
  - Server Based
    - Static location
    - Only exist once
  - Service or User Based
    - Dynamic location
    - Can exist more then once

Identity-Based Cryptography

- Retrieves the setup
- Generate Public Key
- Generate Private Key

- Encrypt the data
- Generate Public Key
- Decrypt the data



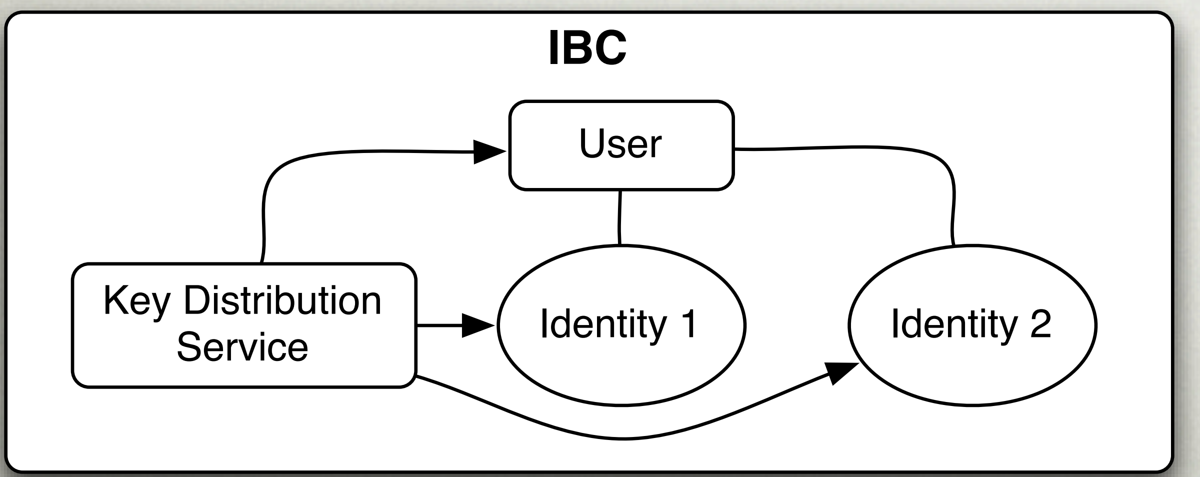
Public Key Infrastructure vs IBC

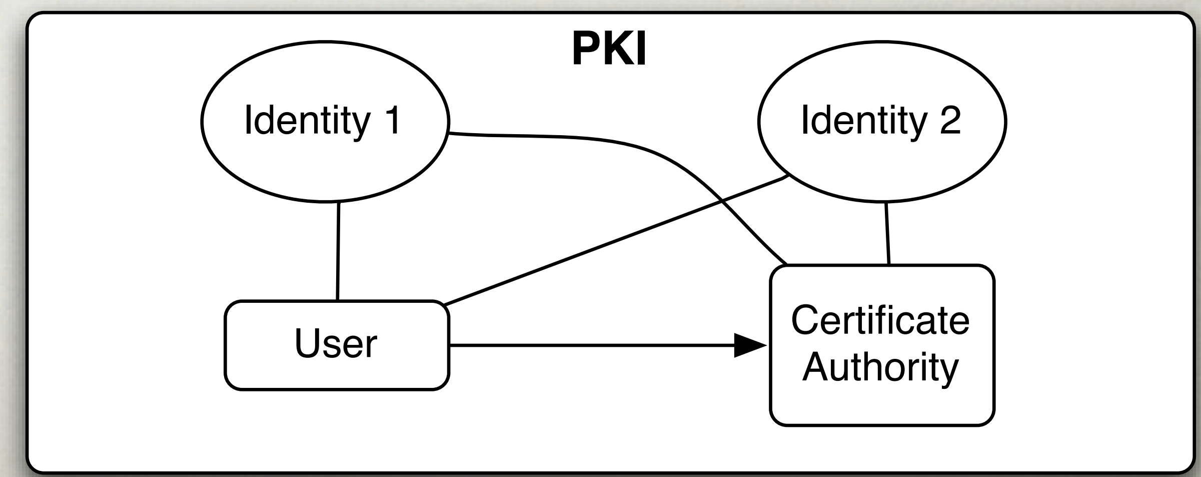
IBC

- Public Key Based on an identity.
- All the keys are generated on the client except the private master key.

PKI

- Public Certificate describes an identity.
- The private key and public certificate is distributed to the client.





Public Key Infrastructure vs IBC

- RSA: Only the right identity can see the data because only the right identity has the right private master key and knows his own identity

- PKI: If a CA says the public certificate could be trusted then it is safe to encrypt data with the given private and public key for the described destination identity

- IBC: Only the identity for which the data was encrypted could understand the data

IBC implementations

- Voltage Identity-Based Encryption
  - Certificate-Based Cryptography
  - Commercial C library

- Eyebee of the University of Ireland
  - Certificate-Based Cryptography
  - Java library

the Eyebee experiment

- Created an Eyebee implementation
- Test Class
- Experiment

the Eyebee experiment

- Created an Eyebee implementation Java Class
  - Generate a Private Master Key
  - Encrypt data by the Private Master Key and the destination identity
  - Decrypt data by the Private Master Key and the destination identity.

the Eyebee experiment

- Test Java Class
  - Create a message: Test Token key #1
  - Generate a Private Master Key
  - Encrypt the message with the identity: Rudy.Borgstede@gmail.com and the Private Master Key
  - Decrypt the message with the identity and the Private Master Key.
  - Print the message in the terminal

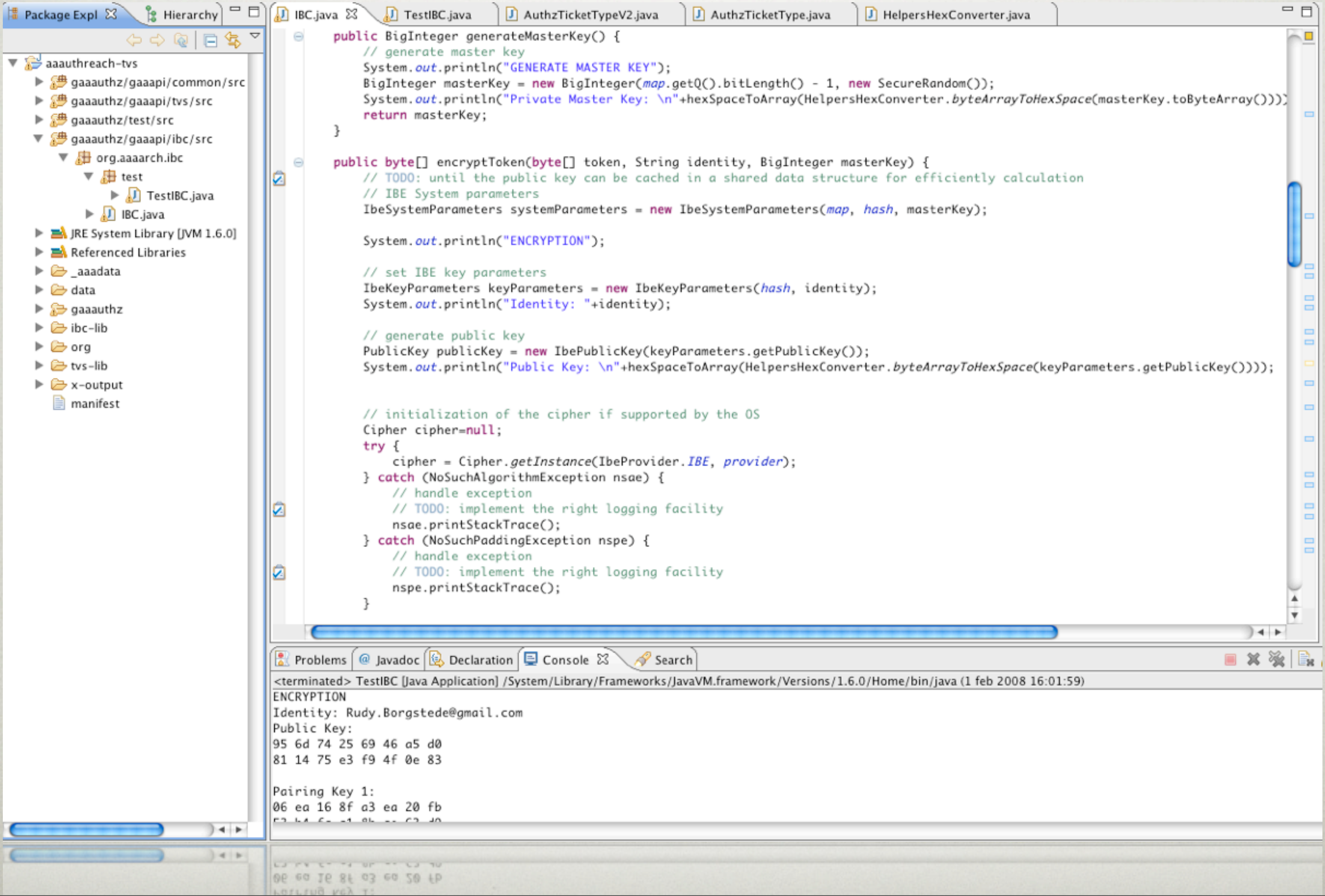
the Eyebee experiment

- Experiment
  - Addepted the implementation class to print the keys, message and identity
  - Test Message: Test Token key #1
  - Identity: Rudy.Borgstede@gmail.com
  - Identity Hash:
    - 95 6d 74 25 69 46 a5 d0 81 14 75 e3 f9 4f 0e 83
  - Private Master Key:
    - 7c 01 fc 3e 86 c6 cf 51 60 c5 d5 95 52 1a c4 5f
    - c1 5e 7d bb 5e 06 6d 19

the Eyebee experiment

- Experiment
  - Public Key with the identity:
    - 03 26 0e 4b 97 9a cb dd b7 9a 57 b7 29 3b cb 26
    - 69 9e c9 75 55 9b e7 45 f9 7a f1 d1 cb 8c 04 1e
    - cb 13 9e 7e 38 99 8b 27 16 c3 a4 8f e6 89 bb ae
    - 52 f9 1f a1 29 bc 20 9b 49 31 da b8 91 a7 8e 4c
  - Private Key
    - 02 a7 86 92 99 d3 61 64 bc f7 17 4c 32 14 64 c1
    - 4c 50 ee 8c 72 2f 1b 07 f5 5f 9c 10 79 5f 82 6f
    - 46 45 1e cf 53 cc ef 51 f6 25 58 19 90 ae 57 1f
    - fc 87 65 cf ec 81 40 db 24 ce 3b e8 a0 7c 39 a7

the Eyebee experiment



```java
public BigInteger generateMasterKey() {
    // generate master key
    System.out.println("GENERATE MASTER KEY");
    BigInteger masterKey = new BigInteger(map.getQ().bitLength() - 1, new SecureRandom());
    System.out.println("Private Master Key: \n"+hexSpaceToArray(HelpersHexConverter.byteArrayToHexSpace(masterKey.toByteArray()))
    return masterKey;
}

public byte[] encryptToken(byte[] token, String identity, BigInteger masterKey) {
    // TODO: until the public key can be cached in a shared data structure for efficiently calculation
    // IBE System parameters
    IbeSystemParameters systemParameters = new IbeSystemParameters(map, hash, masterKey);

    System.out.println("ENCRYPTION");

    // set IBE key parameters
    IbeKeyParameters keyParameters = new IbeKeyParameters(hash, identity);
    System.out.println("Identity: "+identity);

    // generate public key
    PublicKey publicKey = new IbePublicKey(keyParameters.getPublicKey());
    System.out.println("Public Key: \n"+hexSpaceToArray(HelpersHexConverter.byteArrayToHexSpace(keyParameters.getPublicKey())));


    // initialization of the cipher if supported by the OS
    Cipher cipher=null;
    try {
        cipher = Cipher.getInstance(IbeProvider.IBE, provider);
    } catch (NoSuchAlgorithmException nsae) {
        // handle exception
        // TODO: implement the right logging facility
        nsae.printStackTrace();
    } catch (NoSuchPaddingException nspe) {
        // handle exception
        // TODO: implement the right logging facility
        nspe.printStackTrace();
    }
```

```
<terminated> TestIBC [Java Application] /System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Home/bin/java (1 feb 2008 16:01:59)
ENCRYPTION
Identity: Rudy.Borgstede@gmail.com
Public Key:
95 6d 74 25 69 46 a5 d0
81 14 75 e3 f9 4f 0e 83

Pairing Key 1:
06 ea 16 8f a3 ea 20 fb
```

Should we use IBC?

- Not yet in a critical production environment.
  - It hasn’t been extensively tested
  - It isn’t a standard
- The Java aaauthreach project
- It is a better security model

Questions?