Privacy

Privacy-enhanced (&Trust aware) Authz in Constained Environm.

Agenda

Constrained Environments and IoT

Privacy in IoT

What are Credentials, what is authn, authz?

Reasonong about Credentials

How does that fit in IoT ?

Privacy-Enhanced Tokens

Conclusions / Summary

Agenda

Constrained Environments and IoT

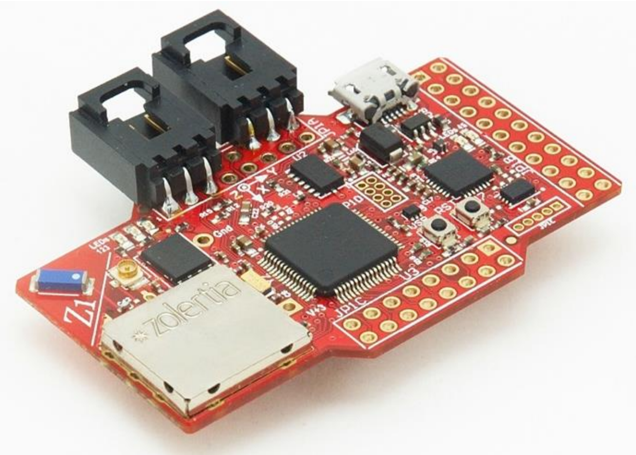
ACE Charter

Standardized solution for authorization delegation

- **use CoAP and leverage DTLS** security where possible
- employ **additional less-constrained devices** in order to relieve the constrained nodes
- **existing** authentication and authorization protocols are used and re-applied ... **restricting** the options within each of the specifications
- operate across **multiple domains**
- **intermittent connectivity of resource server**

Constrained Device?

- Flash Memory say, ~ 512KB, RAM, say ~32KB
- Energy constraints
- No user interface/unattended
- Nodes must sleep often
- LLN: low power, lossy NW
  - ~ 100kb/sec, high loss, high variability
  - Physical layer may be constrained to ~100 bytes/message



CoAP

The Constrained Application Protocol

- implements HTTP's **REST model**
  - GET, PUT, DELETE, POST; media type model
  - while avoiding most of the compl·exities of HTIP

Simple protocol, datagram only (UDP, DTLS)

- 4-byte header, compact yet simple options encoding
  - adds "observe", a lean notification architecture

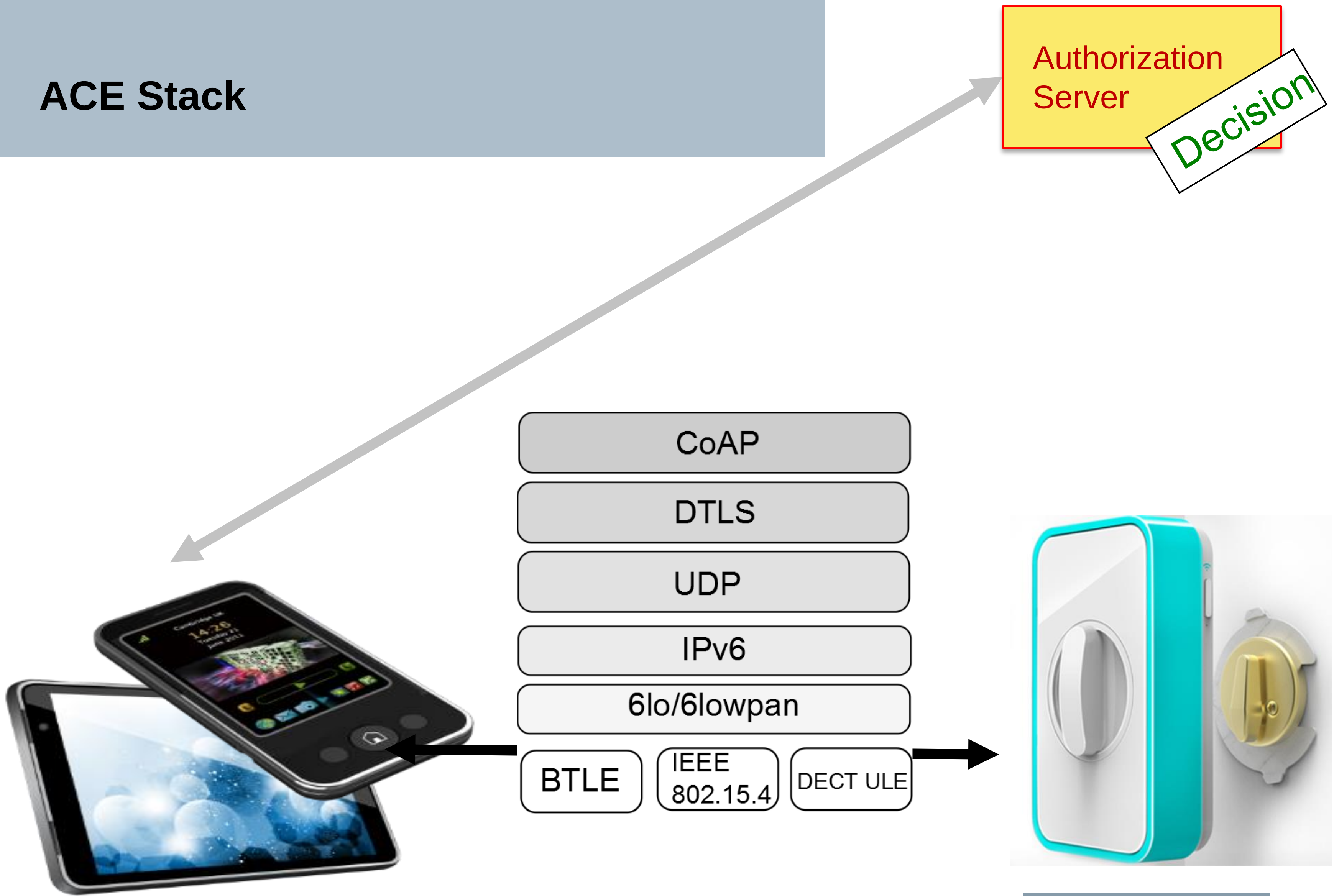
GET coap://temp1 .25b006.floor1 .example.corn/temperature

PUT coap://blue-lights.bu036.floor1 .example.corn/intensity

GET coap://25b006.floor1 .example.com/.well-known/core

</temp>: n="TemperatureC " ,</light>:ct=41;n="LightLux"‘

ACE Stack



Authorization Server

Decision

CoAP

DTLS

UDP

IPv6

6lo/6lowpan

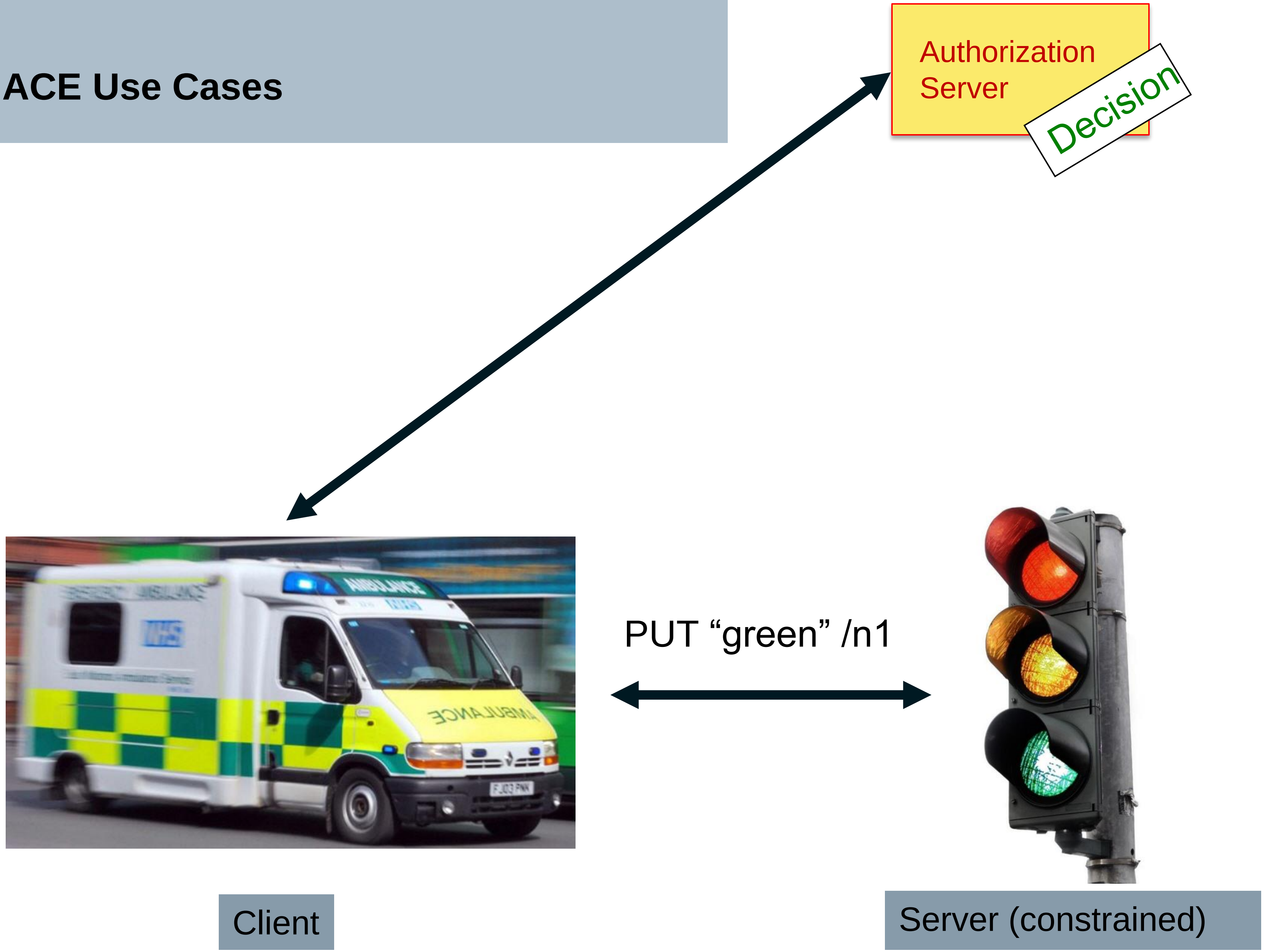
BTLE	IEEE 802.15.4	DECT ULE

Client

Server (constrained)

How to support explicit, dynamic authorization?

ACE Use Cases



Authorization Server

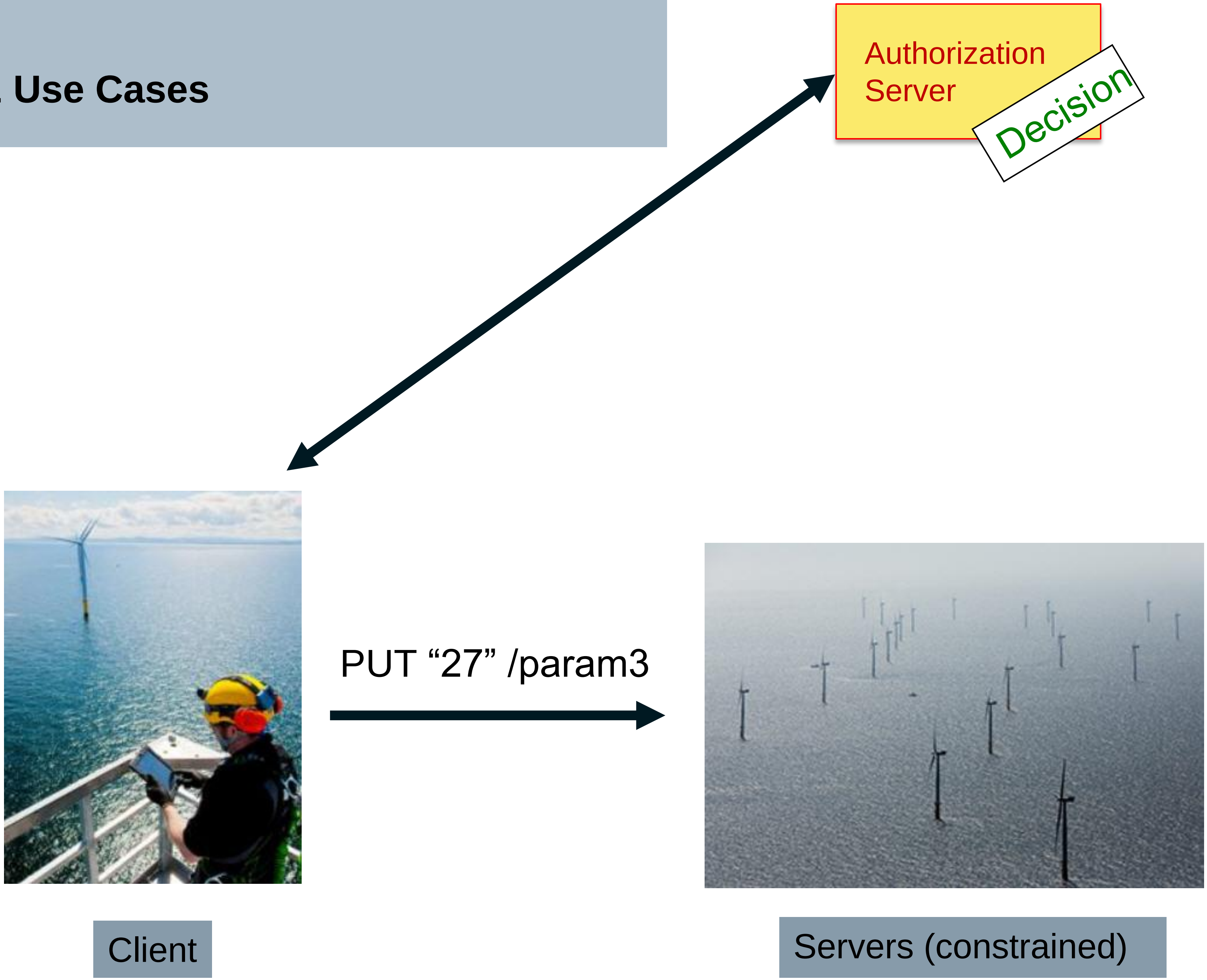
Decision

PUT “green” /n1

Client

Server (constrained)

ACE Use Cases



Authorization Server

Decision

PUT “27” /param3

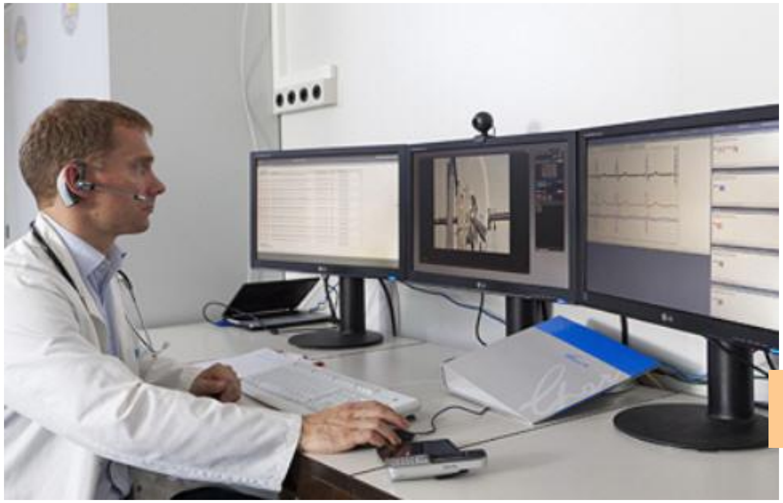
Client

Servers (constrained)

ACE Use Cases

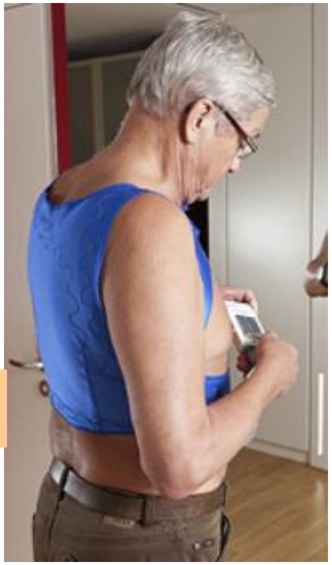
Authorization Server

Decision



GET /bloodpressure

PUT “2.5mg” /sedative



Client

Servers (constrained)

Agenda

Privacy in IoT

DP Officers: Mauritius Declaration

IoT’s sensor data is

- high in quantity, quality, sensitivity
- sensitive inferences that can be drawn
- identifiability is rather likely

IoT data should be regarded & treated as personal data

… huge challenges will be faced by IoT developers, authorities, and individuals

IoT sensor data

Will disclose

- location information
- Relation between people
- Preferences and routine activites

To skript kiddies !

Big Data

Data is an asset

- it generates value for the data controller (processor)

… instead of instructing a computer what do, throw data at the problem and tell the computer to figure it out

    - Kenneth Cukier, editor of “The Economist”

Open data is data that can be freely used, reused and redistributed by anyone

- subject only, at most, to the requirement to attribute and sharealike
    - opendefinition.org

Big Data / IoT vs Privacy

"Barriers against the free flow of data are, in effect,

- barriers against trade"
  - Carl Bildt, former prime minister of Sweden
    chair of Global Commission on Internet Governance

- "DP officers have lost contact with reality"
  - NN

Fitting Policies in IoT-A

Virtual Box

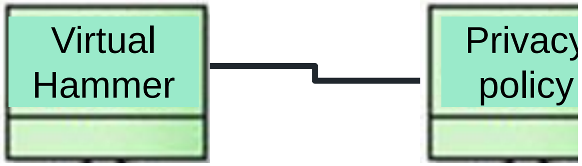
I need a nail !!

This is my hammer
I want rerum to enforce
<these policies>

Virtual Nail

You can use me

Destroy the hammer after 5 days
Don‘t give this hammer to UniDa



Privacy Policies are SW Artefacts
Associated to the Virtual Entity belonging to a Data Subject

When taking decision bout using the associated Data or Service, the policy is enforced

Who decides the policies?
The Data Subject

Pseudonyms are useful

- We require different layers of pseudonyms
  - At least one for “cloud”, one for wireless NWs

- Authorized entities must be able to
  - accept (somehow) pseudonyms
  - without explicit communication to an authority

- Pseudonyms must be compatible with key management

Agenda

What are Credentials, What is authn, authz?

- Well-known definitions

Well-known definitions: Authentication

RFC2828 Internet Security Glossary, 2000

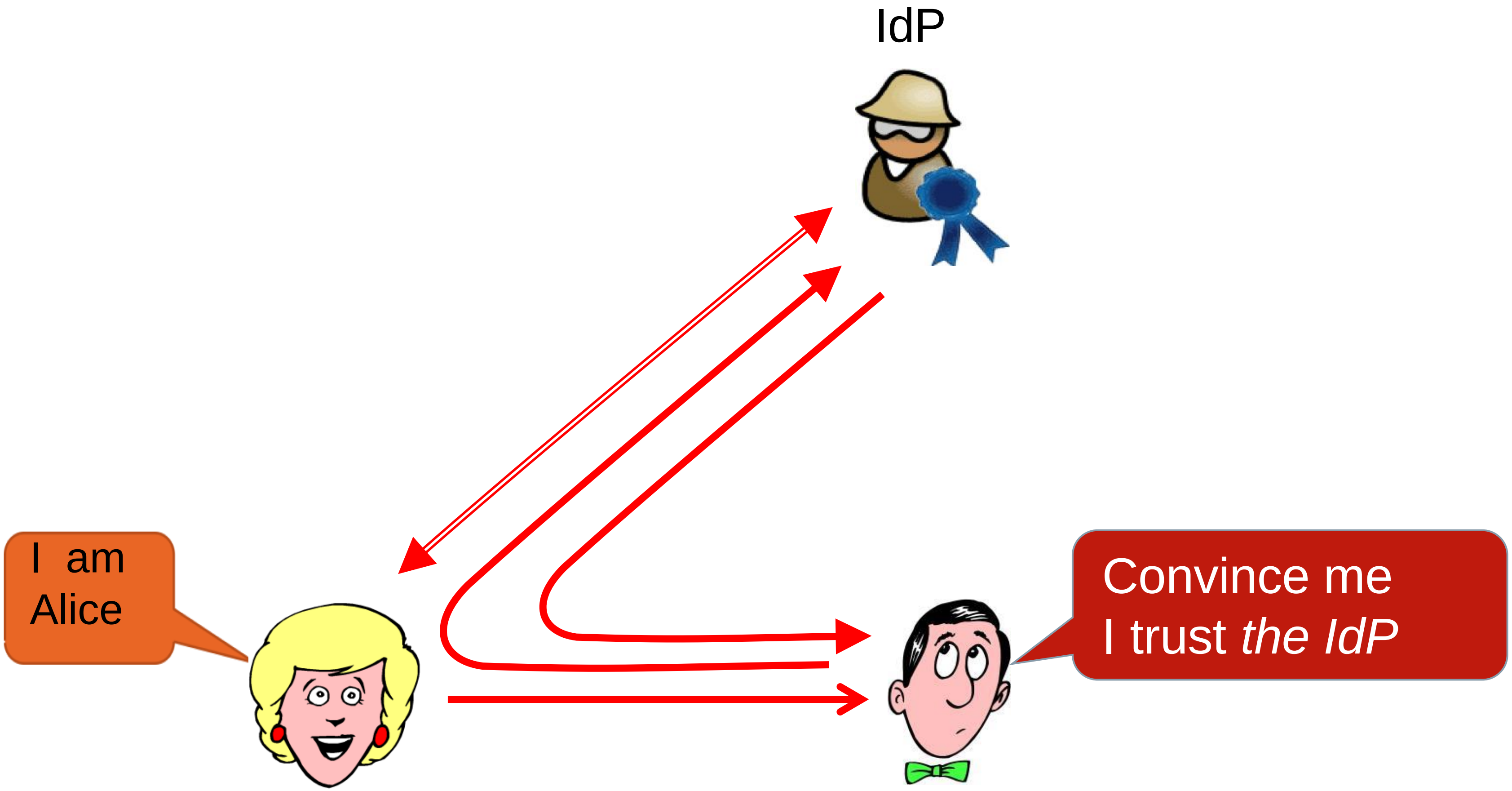
The process of verifying (i.e., establish the truth of) an identity claimed by or for a system entity

consists of two steps:

1. Identification step: Presenting an identifier to the security system
   - Identification: An act or process that presents an identifier to a system so that the system can recognize a system entity and distinguish it from other entities
   - Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control

2. Verification step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

IdP (say: SAML)



IdP

I am Alice

Convince me
I trust *the IdP*

Authentication

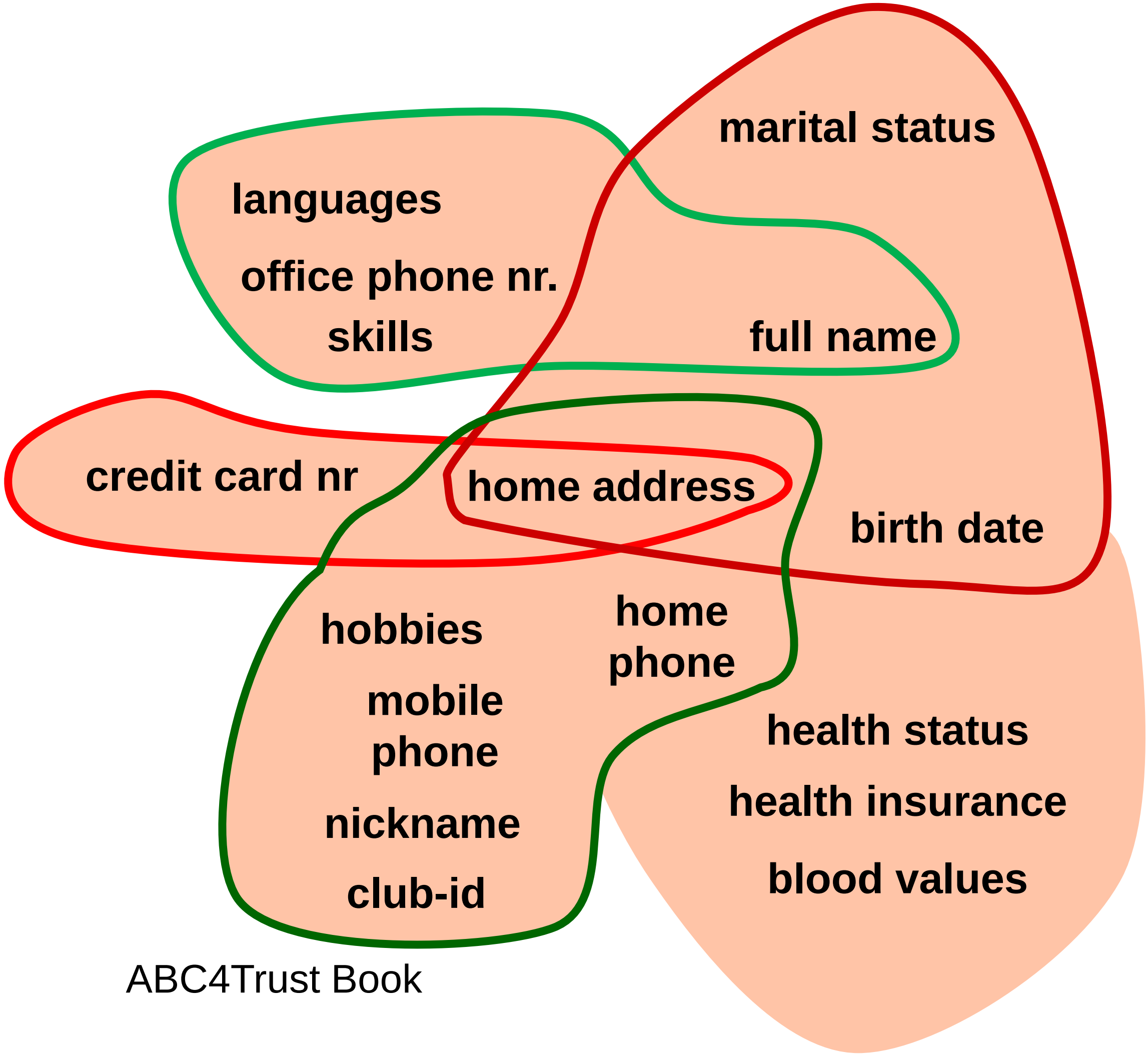
Identification is often first step of a transaction

I am Alice
Is she?

Makes sense in an organizational environment ... but

- People have several different names (or nicknames)
  - used in different contexts (students card; Club ID, drivers lic.)
- *All* transactions from *all* different contexts are linkable
  - The SAML IdP knows quite a bit of yourself
- Not reasonable to show all attributes on each transaction

Identity (Partial Identity): Set of attributes related to an entity in a certain context



ABC4Trust Book

Bad definitions: Authentication

RFC2828 Internet Security Glossary, 2000

The process of verifying (i.e., establish the truth of) an identity claimed by or for a system entity

~~Consists of two steps~~:

1. ~~Identification step: Presenting an identifie~~r to the security system

- Identification: An act or process that presents an identifier to a system so that the system can ~~recognize a system entity and distinguish it from other entities~~
- ~~Identifiers~~ should be assigned carefully, ✓ ause authenticated identities ~~are the basis for other security services, such as access control~~

2. Verification step: Presenting or generating authentication information that corroborates the binding between the ~~entity and the identifier~~

Well-known definitions:
Credential

The typical answer: It is either

1 Something you have

- Security tokens
- Smart cards
- Money (is *that* a credential?)

2 Something you are

- Biometrics
  - Signature dynamics
  - Keyboard dynamics
  - Voice print

3 Something you know

- Passwords
- Passphrases
- Shared secrets (e.g. mother's maiden name)
- How to solve a (set of) problems (puzzles)

Bad definitions: Credential

The typical answer: It is either

~~1 Something you have~~
~~- Security tokens~~
~~- Smart cards~~
~~- Money (is *that* a credential?)~~

~~2 Something you are~~
~~- Biometrics~~
~~- Signature dynamics~~
~~- Keyboard dynamics~~
~~- Voice print~~

~~3 Something you know~~
~~- Passwords~~
~~- Passphrases~~
~~- Shared secrets (e.g. mother's maiden name)~~
~~- How to solve a (set of) problems (puzzles)~~

Too complex:

- We want to *reason* about credentials
- In a simple and coherent way

Bad usage:
Crypto

How expensive is crypto

Could you encrypt (in IoT) 3 bits using 3 bits?

No: padding

No: TLS, DTLS

No: randomization is necessary

No: flags

Agenda

Reasonong about Credentials
What are Credentials?
How do you reason about them & policies?

My Definitions

Credential:

Is a claim endorsed by ***somebody***

- That binds an ***attribute*** *(or **predicate** on attributes)* to a *(set of)* ***problems***

Examples:

Problem	Credential
Providing the correct password or PIN	PW DB
Responding a "public key" – based challenge whose solution is verifiable using the public key	PKI Cert
Providing money	Bank Note
Having a face that matches a certain photo	Passport / Univ ID
ZKP	ZKP Certs

My Definitions

Credentials may be revoked in several ways,

- for instance money gets immediately revoked (or changes the “subject”) as soon as it is used

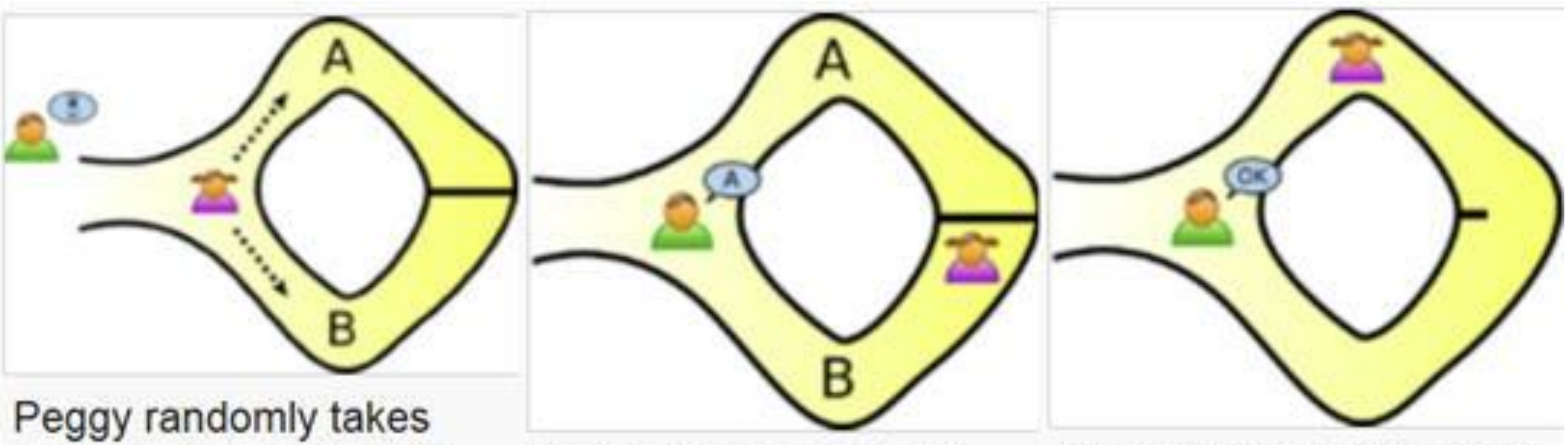
- **Problems** and **credentials** can be used to construct **secure channels**
- which provide some security goals,
  - like authenticity or integrity, non-repudiation, etc
  - to one or both of the communication partners while
- assuring that the other partner has some attributes

What is a credential?

Is *this* a credential?

Ali Baba is the only one who can open the door

Peggy wants to prove that she is Ali Baba

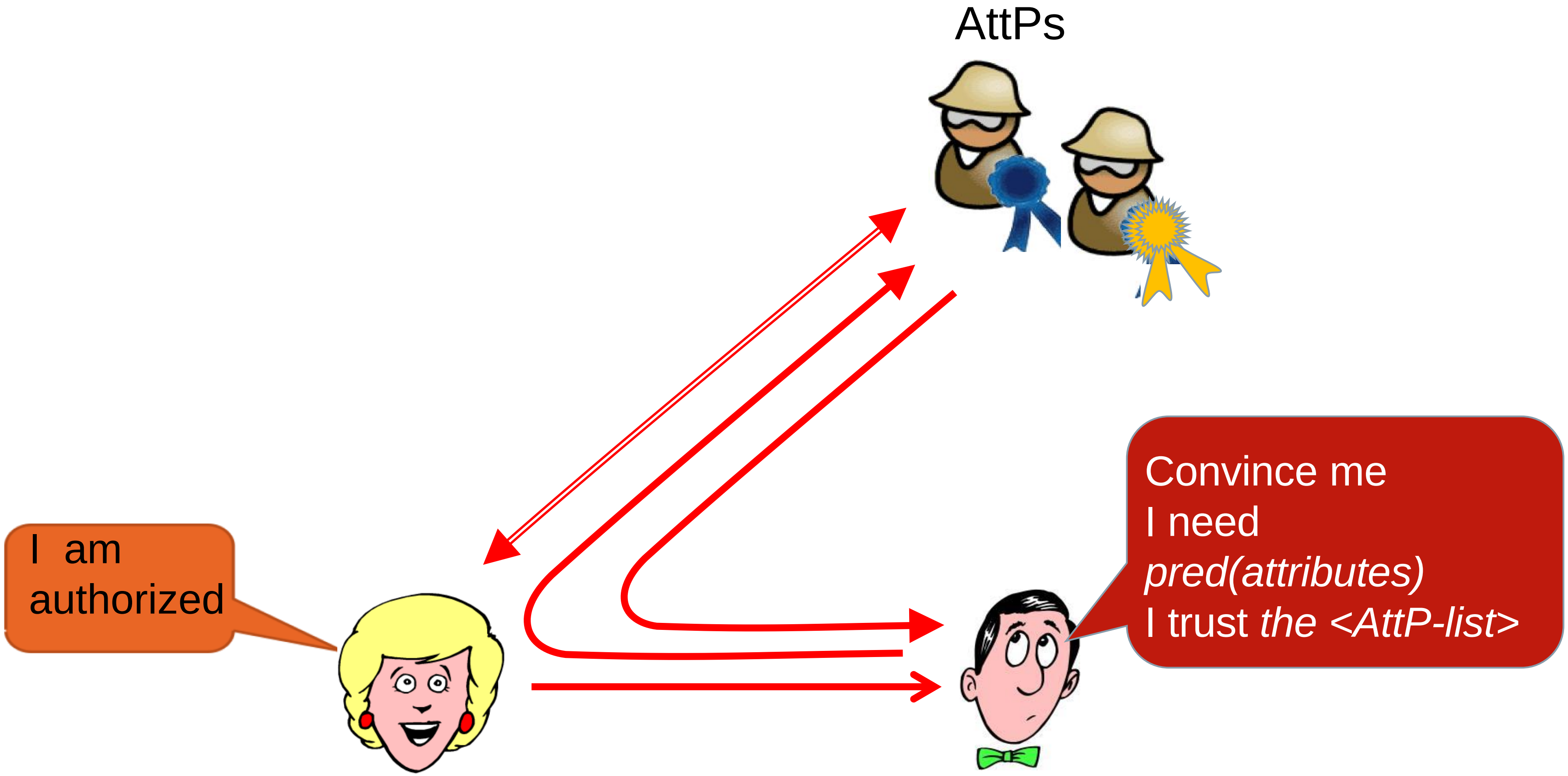


Peggy randomly takes either path A or B, while Victor waits outside

Victor chooses an exit path

Peggy reliably appears at the exit Victor names

Attribute-Based „Authentication“ / Authorization



My Definitions

Attributes:

may be seen as pairs: attribute type and value

but may contain other “fields” for “admin domain” / “context” / “validity”

I tend to think of values as ordered, say in a lattice

Privacy

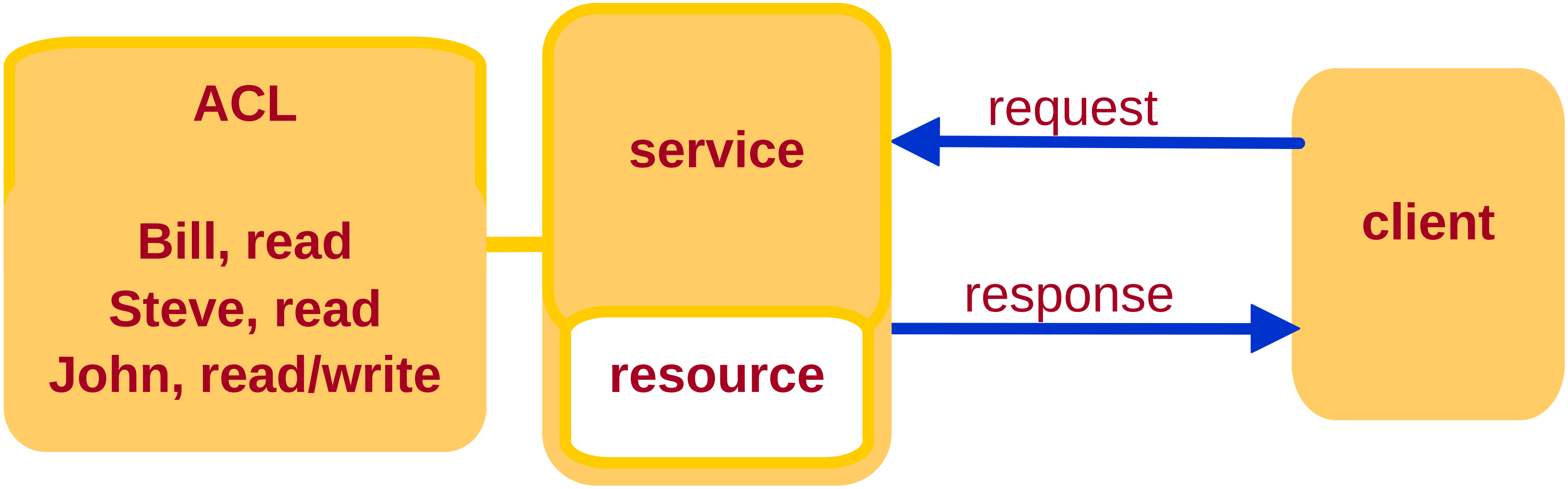
Protection of user's privacy

- unlinkability (multi-use)
- using/combining multiple credentials
- selective disclosure of credentials (or attributes)
- predicated over attributes

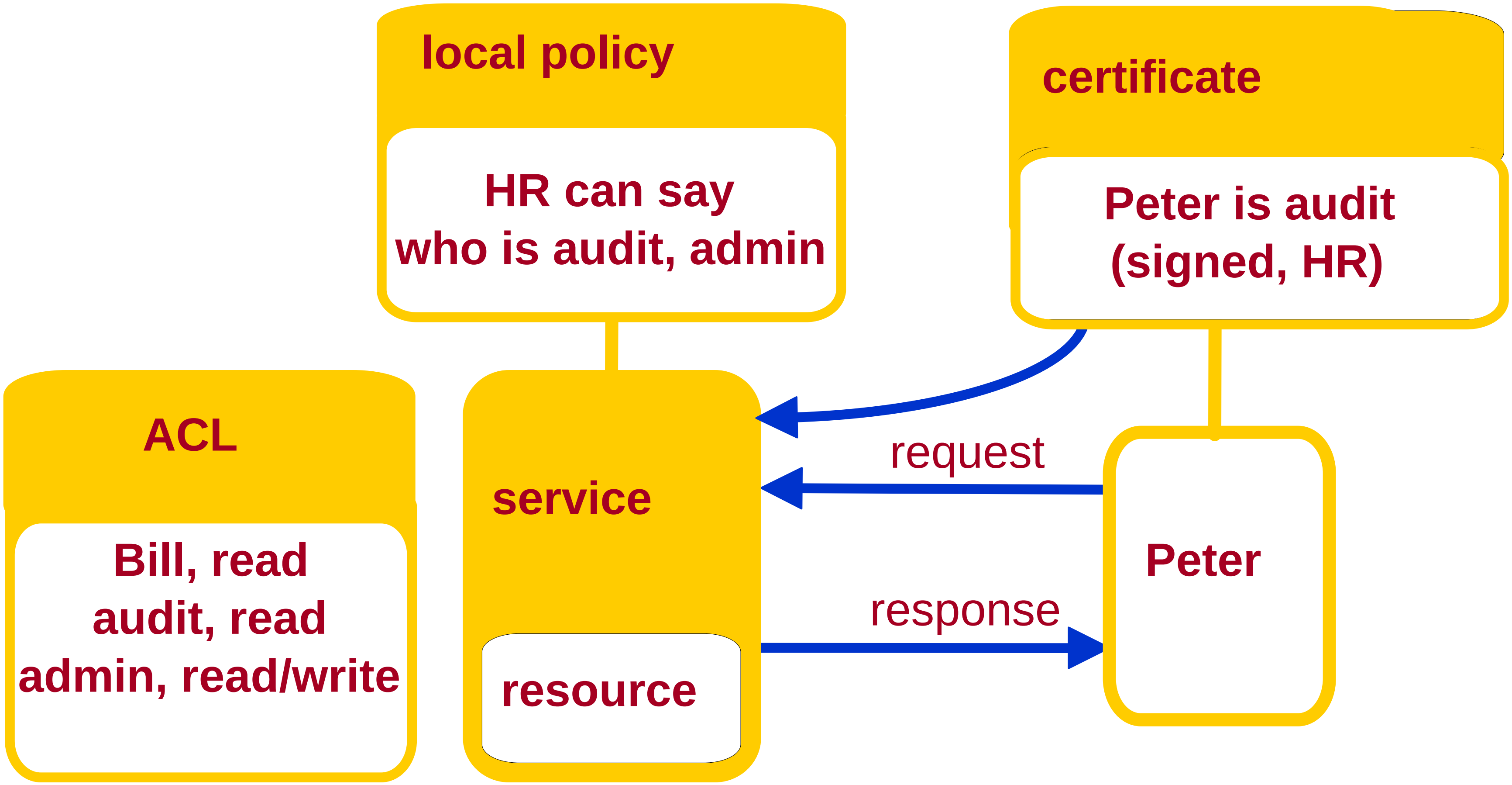
Strong authentication

- unforgeability of presentation tokens
  - Nobody should not be able to show a token for a credential that she never obtained

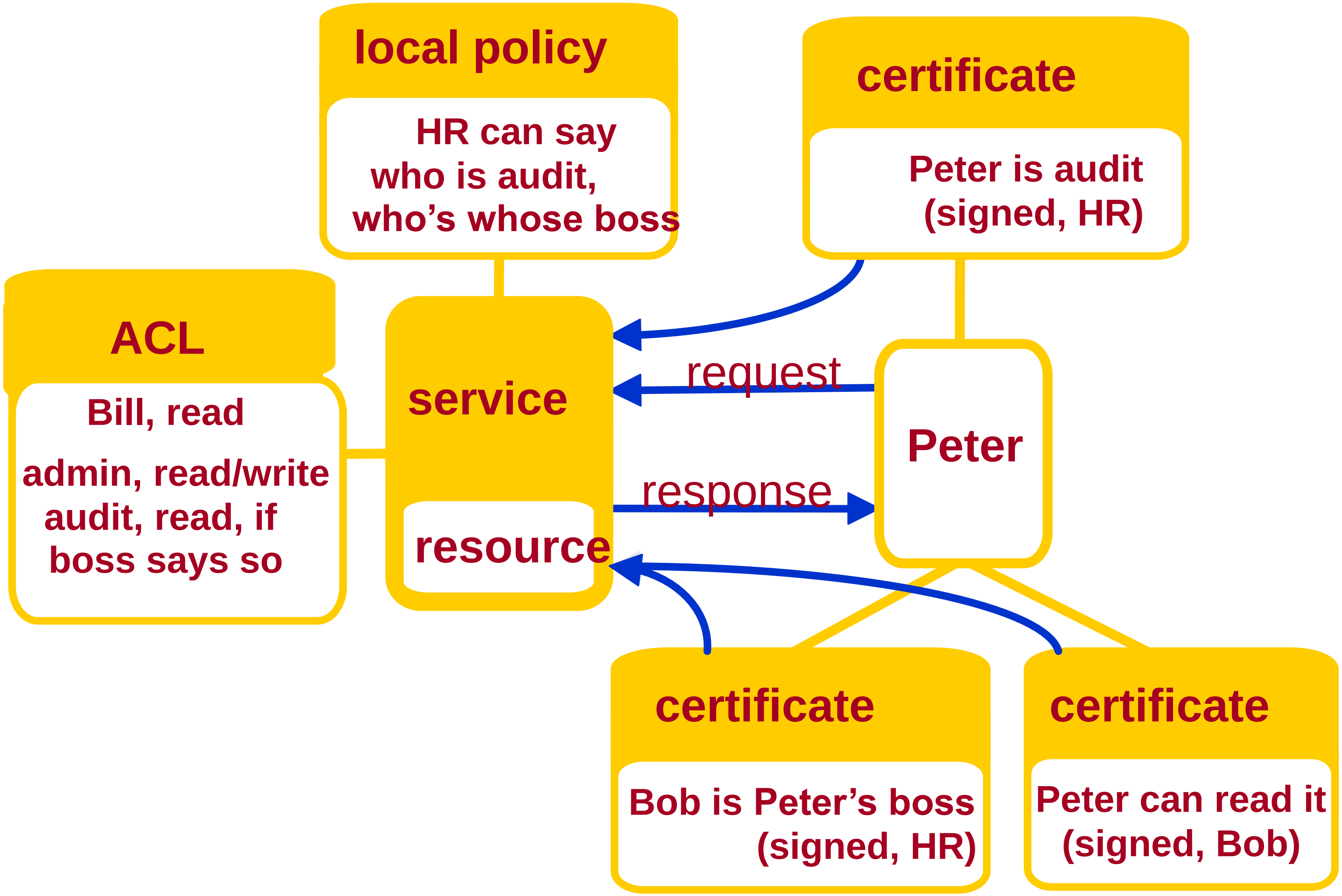
Simple Example



more complex Example



Even more complex Example


local policy
HR can say who is audit, who’s whose boss
certificate
Peter is audit (signed, HR)
ACL
Bill, read
admin, read/write
audit, read, if boss says so
service
resource
request
response
Peter
certificate
Bob is Peter’s boss (signed, HR)
certificate
Peter can read it (signed, Bob)

Composition

$$\frac{A \ \overset{c}{\text{—}\bullet}\ B \qquad\qquad A \ \overset{PW}{\text{——}>}\ B}{A \bullet \overset{c}{\text{—}\bullet}\ B}$$

$$\frac{A \ \text{—}\bullet\ SC \qquad\qquad B \bullet \text{—}SC}{A \ \text{—}\bullet\ B}$$

Composition

$$\frac{A \mathrel{—\!\bullet} \mathrm{Aut} \qquad\qquad B \mathrel{\bullet\, —} \mathrm{Aut}}{A \mathrel{—\!\bullet} B}$$

Aut: something like the RSA token or the Gauthenticator

Q: How to create “multi-domain” Aut and bind them dynamically?

How to reason about 2-level authn?

Agenda

Privacy-Enhanced Tokens

Abstract View

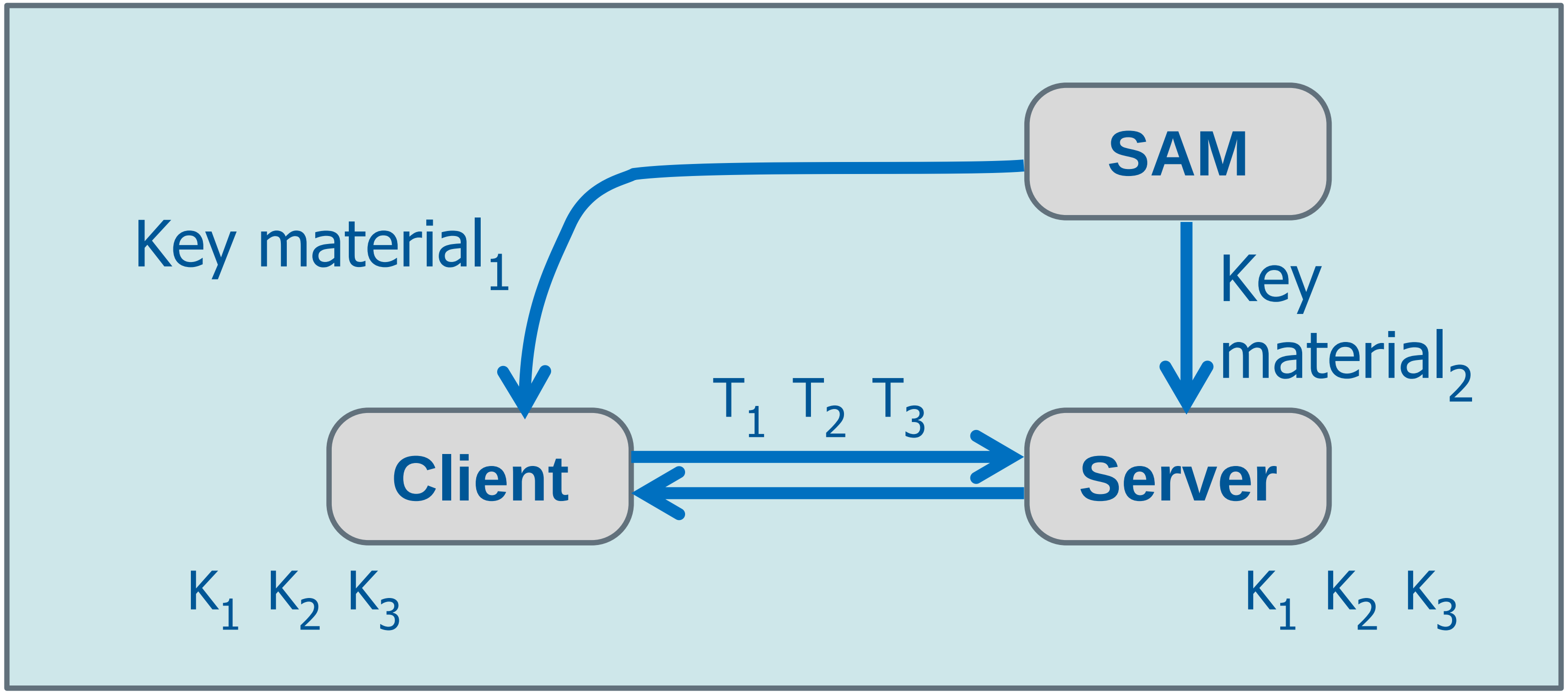
Goals

In some cases Privacy is not an issue

In some cases, Client gets one response per request

- in others, Client subscribes to a stream

In some cases DoS resilience only under stress…



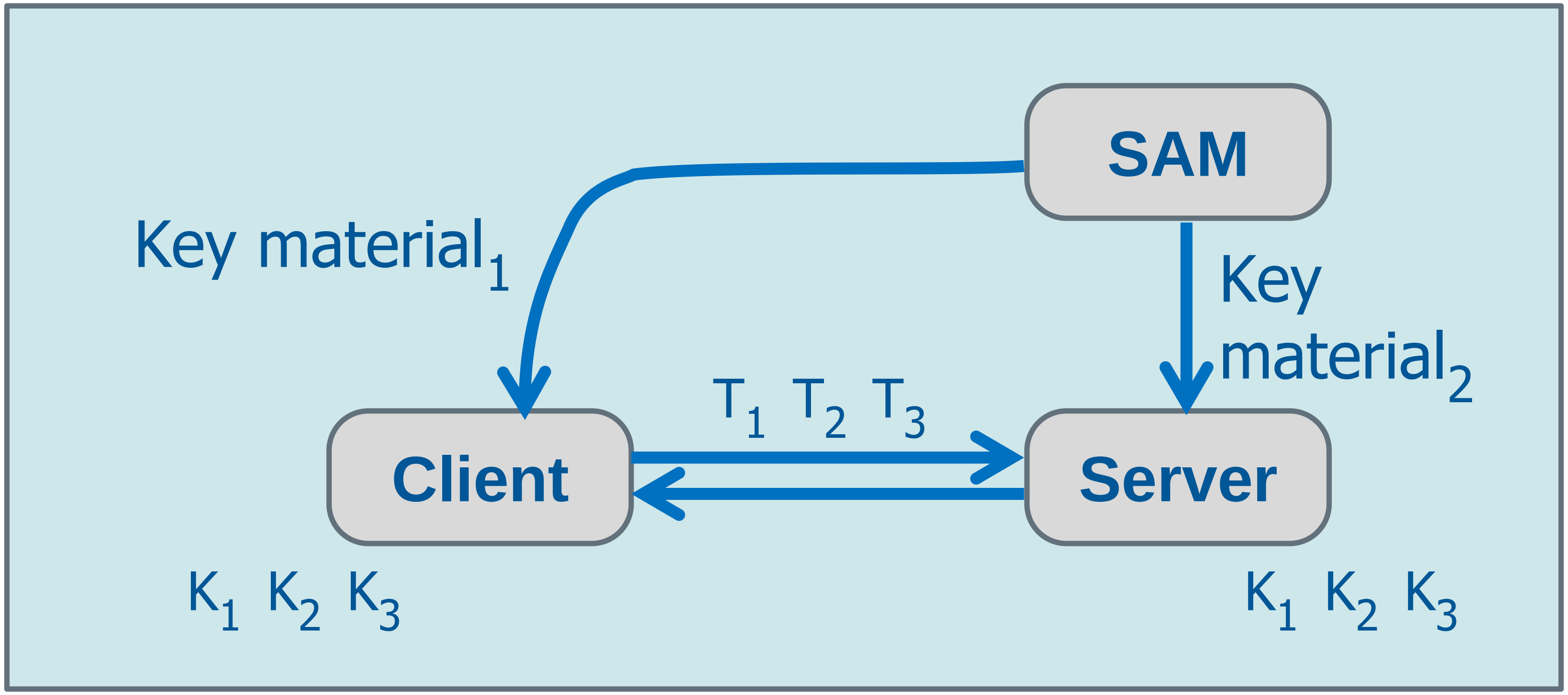
One solution possibly does not fit all

The Key Material allows Client and Server to …

- generate Tokens & keys, verify Tokens

… Many ways of constructing & using tokens/keys

- As one-time-pads
- For DTLS, AES/MACs



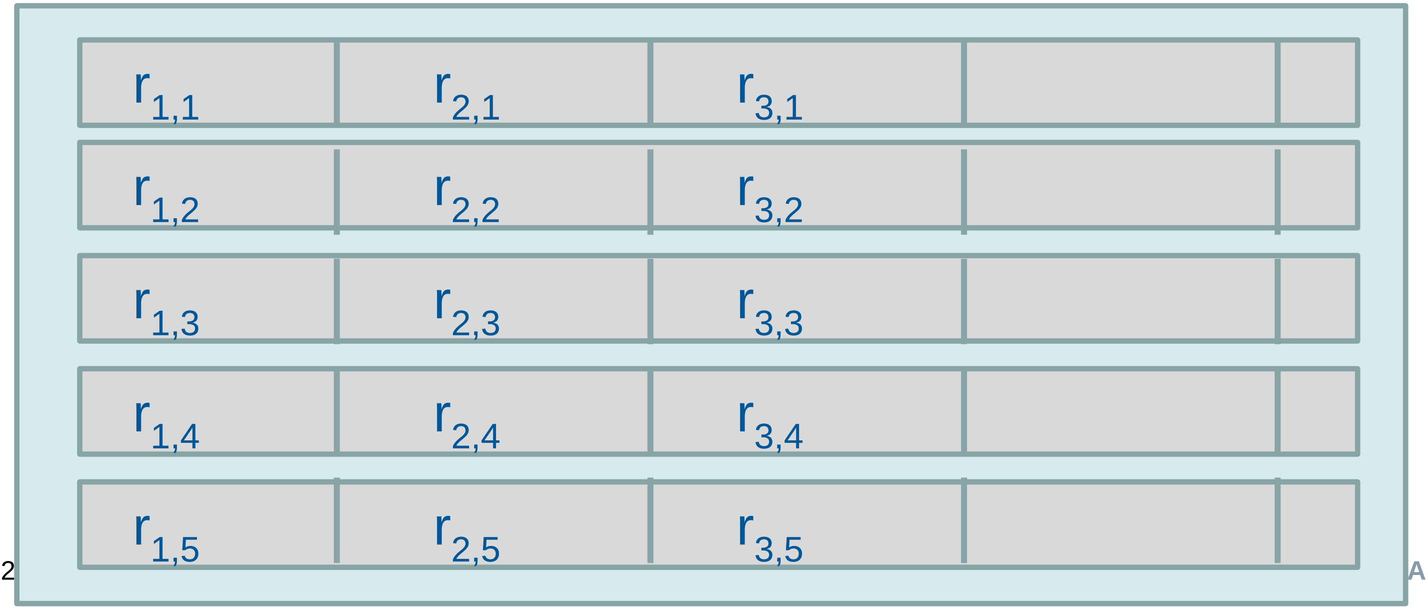
A Low-Cost Solution

Use Pseudo-Random Generators

An attacker may not distinguish if a (long) bit stream

- is purely random
- has been generated by a Pseudo-Random Generator G(k)
  - where k is a (“small”: 128, 256 bits) random key

Let G(k) be written as an array (matrix) of seemingly random bits:



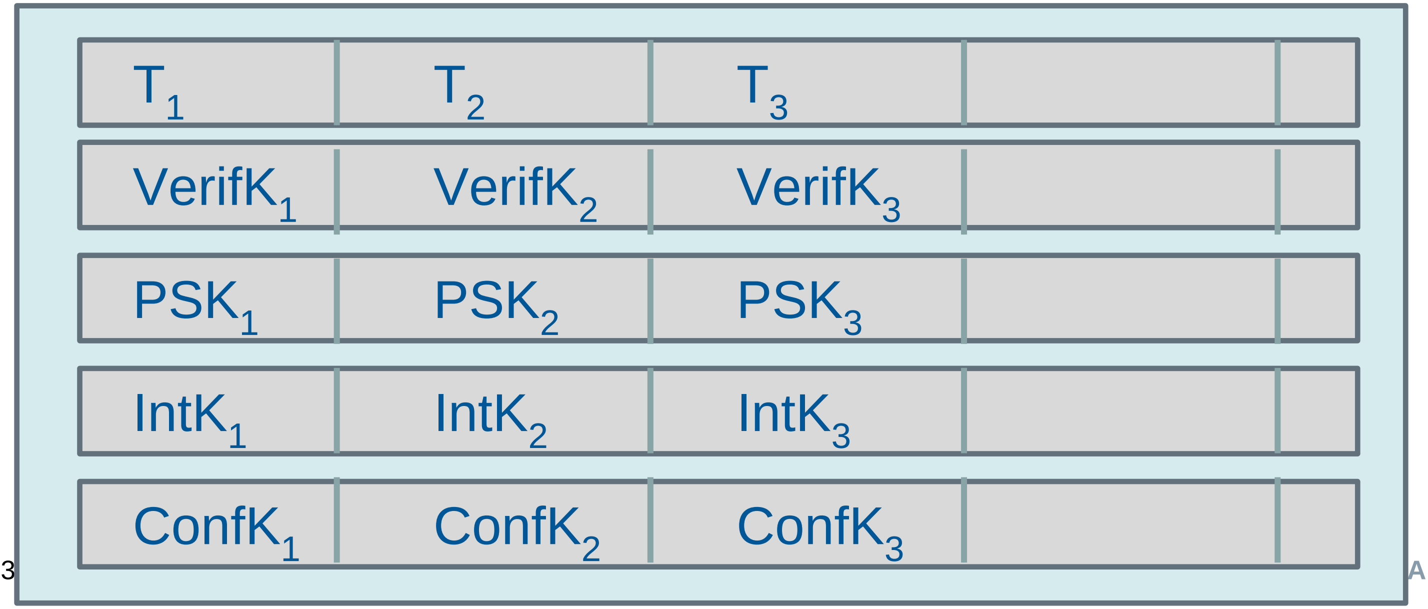
$r_{1,1}$	$r_{2,1}$	$r_{3,1}$		
$r_{1,2}$	$r_{2,2}$	$r_{3,2}$		
$r_{1,3}$	$r_{2,3}$	$r_{3,3}$		
$r_{1,4}$	$r_{2,4}$	$r_{3,4}$		
$r_{1,5}$	$r_{2,5}$	$r_{3,5}$		

A Low-Cost Solution

Not only generate Tokens T1 , T2 … but also …

- Verification Keys (“Proof of Possession”): VerifK1 VerifK2
- Pre-Shared Keys (for DTLS, if required): PSK1 PSK2
- Integrity Keys: IntK1 IntK2
- Confidentiality Keys (for encryption): ConfK1 ConfK2

Use the long pseudo-random stream as a set of “Tokens and keys”



T_1	T_2	T_3		
$VerifK_1$	$VerifK_2$	$VerifK_3$		
PSK_1	PSK_2	PSK_3		
$IntK_1$	$IntK_2$	$IntK_3$		
$ConfK_1$	$ConfK_2$	$ConfK_3$		

SIEMENS

A Low-Cost Solution

Propose to Use ChaCha20

… (or ChaCha7?) as a pseudo-random generator

Use One-Time Pads for Confidentiality

- No need for padding
- Small message sizes

Open for further discussion

- Integrity
  - Propose: publish hashes (not trivial)

SIEMENS

Why ChaCha20 (or ChaCha7)?

Better security, better performance,, saves NW bandwidth

Better security

- ChaCha20 is very simple
  - even a completely naive implementation will be secure
- immune to padding-oracle attacks
  - which affect CBC mode as used in TLS
- immune to timing attacks

Better performance on mobile and wearable devices

- AES-128-GCM, AES-NI disabled: 131 MB/s
- ChaCha20+Poly1305, -march=native: 560 MB/s

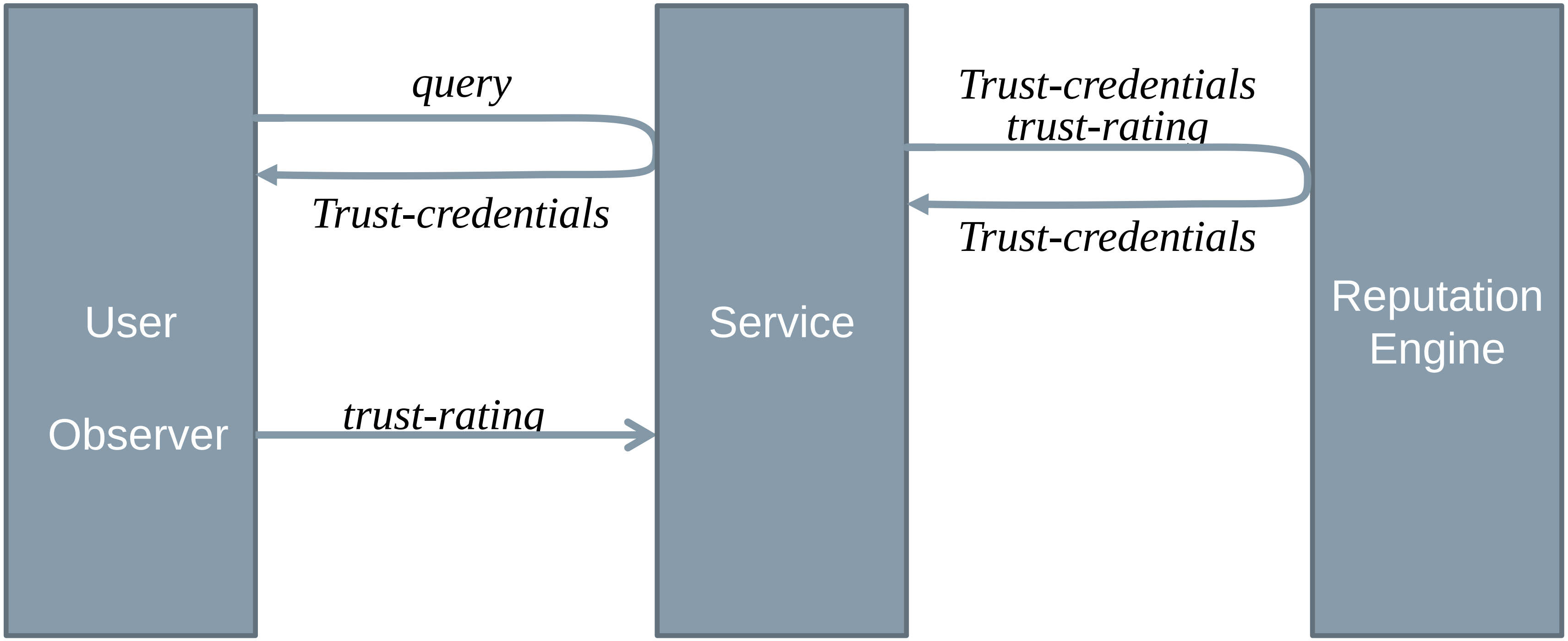
Saves network bandwidth

- Poly1305 (16 bytes) vs HMAC-SHA1 (20 bytes)

Agenda

Trusting Sensors and IoT Services

Trust Management (?)



- Trust of observers?
- Aggregation?
- Assurance via Altcoins (?)

Agenda

Conclusions / Summary

Conclusions

Need to reason about certificates and policies

- … different types of certs, for different purposes

Need to reason about composability

Trust based on <some kind of> certification (certificates)

Summary / Conclusions

I like to see **credentials** as assertions produced/endorsed/written by **somebody** (with some **attributes**) that bind

- sets of **problems** with

- **attributes**

Moreover entities have "local **policies**"

- that say who is able to "say" what types of assertions about what type of people. The author of the credentials may be "authenticated" via attributes, not necessarily identities.

Summary / Conclusions

We will probably need a

- **constructive** approach to channels, credentials, policies…
    - When does the combination of two subprotocols (or channels) provide a solution to a (larger) problem?
    - What are the right logics for reasoning about channels, credentials, policies?

We do not have to solve this “abstract” problem in general, but

- in practical, even simple, applications for constraint devices
- where the **devices have to reason** about credentials / assertions / policies in order to **plug-and-play**

Trust that a system will protect my Privacy

Incentives?

- We need regulation, clear contracts, clear definitions, compliance tools

Perception?

- We need PETs that make privacy more visible and the implementation of privacy rules more transparent

Mass data collection increases the complexity of securing the system

- We need Authz/Consent systems supporting strong Ψnyms
- We need privacy enhancing data sharing / data publishing