



The Leading-Edge Software Solutions

Managing Access Control for Things: a Capability Based Approach

*BODYNETS 2012 - SeTTIT International Workshop
Oslo (Norway)*

September, 24th – 26th, 2012

Domenico Rotondi, Salvatore Piccione
(domenico.rotondi@txtgroup.com)
TXT e-solutions SpA

IoT@Work Project Summary Sheet



■ EU funded Project

- Duration: 3 years
- Approx. 3.5 Mio € Funding
- (Total Budget ~ 5.9 Mio €)
- Started on June 2010
- 6 Partners from Industry and Research

■ Vision

- IoT Supporting Agile & Smart Manufacturing Systems
- Plug&Play IoT solution for Manufacturing

SIEMENS

Project coordinator
network & security

Microsoft | Innovation Center
Europe

Security
Configuration Management



Scenarios & requirements
pilot

TXTe-solutions

Software Engineering &
middleware aspects



Software Engineering
system modelling

inIT Institut
Industrial IT
www.init-owl.de

Automation & Engineering

IoT Peculiarities (Access Control point of view)



- Potentially unbounded number of interacting **subjects** (things, applications, humans)
- Users/usage:
 - ✓ **Non-ICT** skilled users
 - ✓ **Everyday life**
- Interaction patterns:
 - ✓ Traditional patterns: **planned** and **long-lived**
 - ✓ IoT world: **short-lived**, often **casual** and/or **spontaneous**
- Context relevance:
 - ✓ Actions/requests/data/... analysis can depend from a set of other data sources (i.e. requestor/provider context)
- Resource constraints
- ...

Access Control solutions that:

- face the IoT scalability challenge
- are easy to use, easy to understand and easy to manage
- can be deployed on *simple* devices (e.g.: reduce the amount of *supporting* data, communications, etc.)
- are secure
- support advanced features (e.g.: access rights delegation, auditability, ...)
- are flexible
- ...

■ Traditional Access Control models:

✓ RBAC (Role Based Access Control)

- Identities
- Roles
- Identities → Roles assignment
- Trust of Identity Providers (IdP) and/or Service Providers (SP)

✓ ABAC (Attributes Based Access Control)

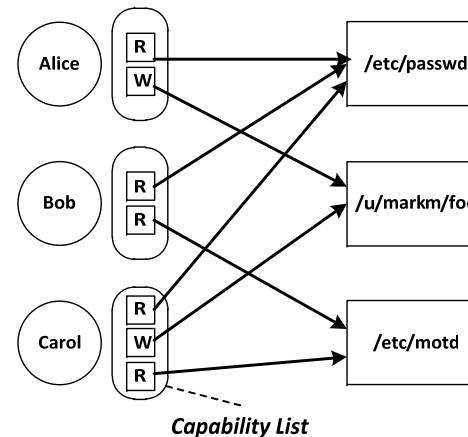
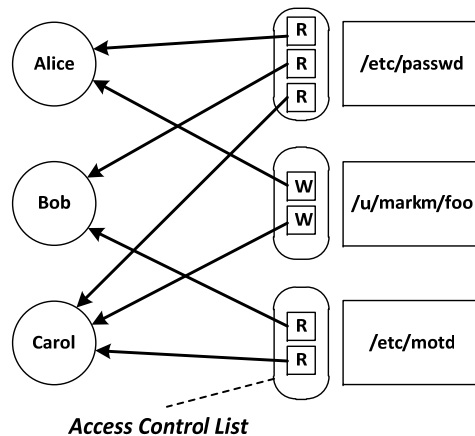
- Attribute names
- Attribute meanings
- Identities
- Trust of IdPs, SPs, Attributes Providers (APs)

■ Issues:

- ✓ Usability (in particular in end-user' centered scenarios)
- ✓ Do not scale **Scalability is a key issue in IoT contexts (explosion of resources/subjects)**
- ✓ Require significant Management effort **Management a nightmare in IoT contexts (explosion of resources/operations)**
- ✓ Identity/Right delegation is complex **IoT can require complex and efficient delegation chains (many more services to orchestrate/integrate)**
- ✓ Security issues, auditability

■ Capability based security: what is it?

- is a security model in which “... *a capability (known in some systems as a key) is a communicable, unforgeable token of authority. It refers to a value that references an object along with an associated set of access rights*”



■ Not a new concept:

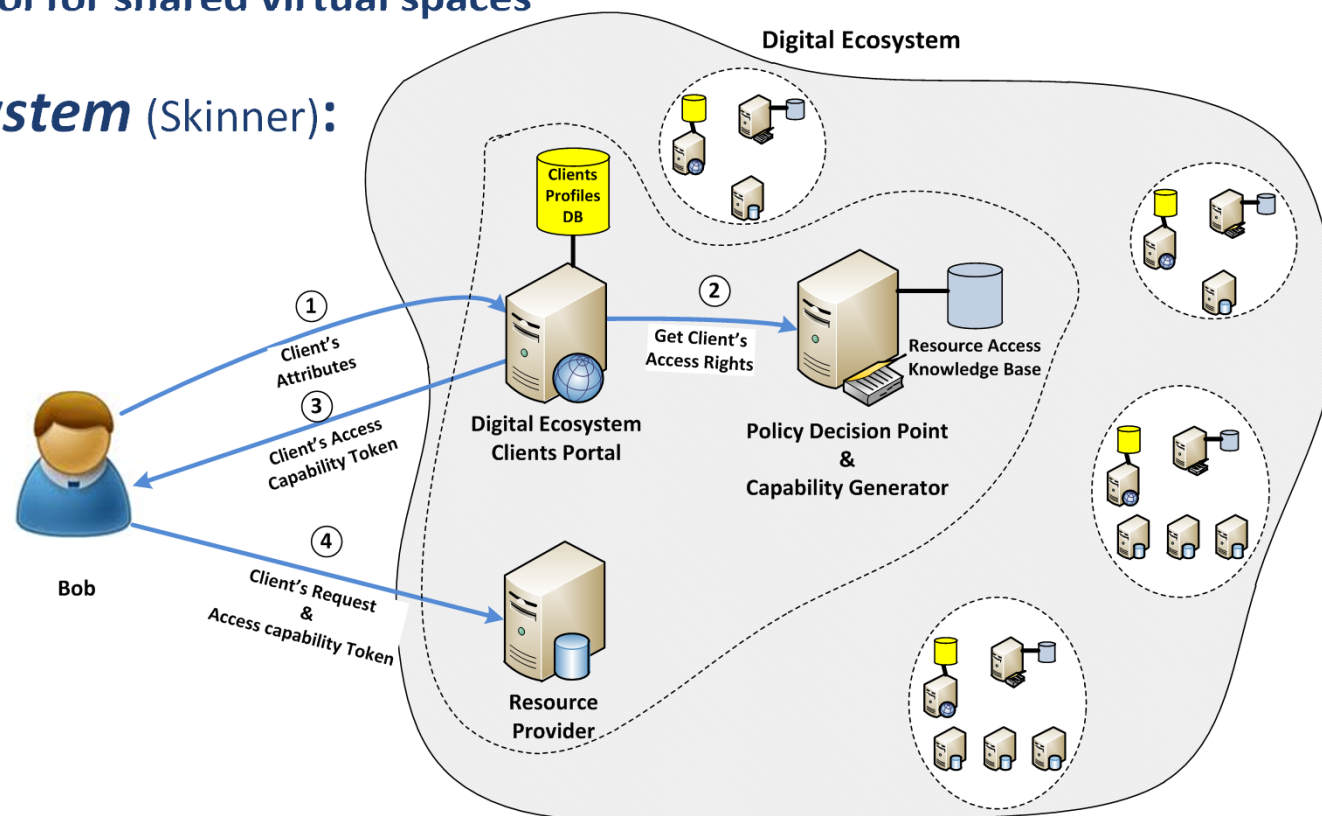
- Levy “*Capability-Based Computer Systems*” (1984)
- Tanenbaum “*Using Sparse Capabilities in a Distributed Operating System*” (1986)
- “RFC2693 - SPKI Certificate Theory” (1999)
- Miller “*Capability Myths Demolished*” (2003)
- Karp “*Solving the Transitive Access Problem for the Services Oriented Architecture*” (2010)
- ...

Capab. Based Security: Previous Experiences

- **Xerox Parc Casca Collaboration application:**

- Access control for shared virtual spaces

- **Digital Ecosystem** (Skinner):



- **IBM Trusted Virtual Data Center** (M. Factor):

- **Capability-based Command Security (CbCS)**: extension to the SCSI protocol for access control to networked storage devices

■ Capability Based Authorization Pros:

- ✓ Principle of Least Authority (PoLA) (Least Privilege) is the default
- ✓ Easy to understand and easy to use for end-users
- ✓ more fine-grained access control
- ✓ less security issues (e.g. no *Confused Deputy* problem)
- ✓ capability model externalizes the authorization management process
- ✓ no need to manage issues related to complexity and dynamics of subject's identities

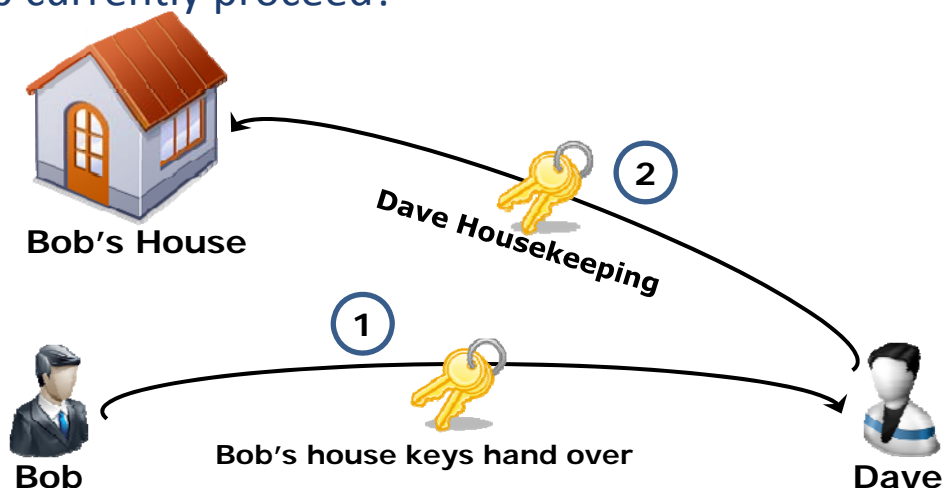
■ Why Capability in FP7 IoT@Work (a factory automation focused project):

- ✓ many subjects (suppliers, maintainers, etc.) need to access resources in the production plant
- ✓ Least Privilege is a must!
- ✓ need to easily delegate rights and to have full auditability of resource access
- ✓ need to offload management to face external subjects dynamics

Use-case Example: current approach

■ The Problem:

- ✓ Bob has to go on holidays
- ✓ his house needs some housekeeping while Bob is away
- ✓ his neighbour Dave offered to take care of Bob's house housekeeping
- ✓ how does Bob currently proceed?

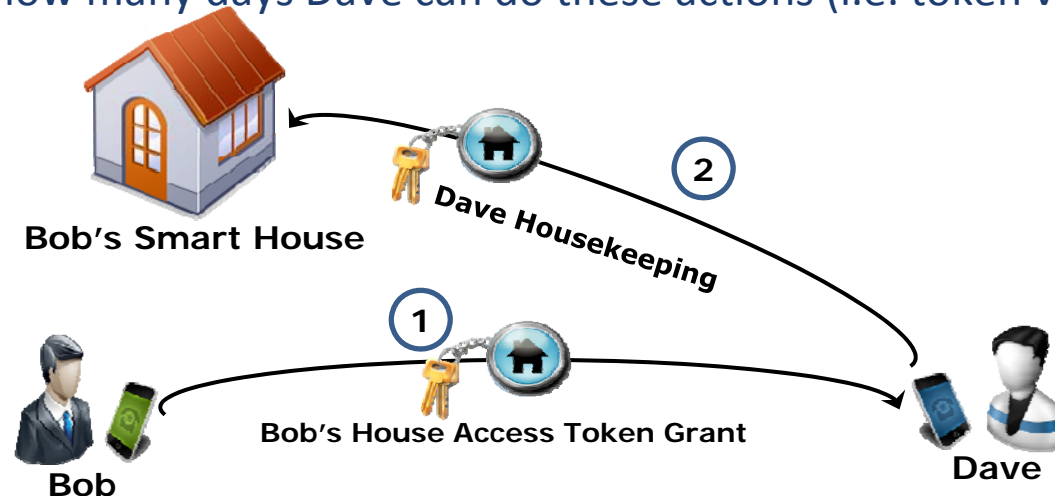


■ Issues:

- ✓ Dave could use the house's keys for non-envisaged/non-authorized activities
- ✓ Dave could make a duplicate of Bob's keys
- ✓ Bob has no real control on what Dave is doing or could do

Use-case Example: capability based approach

- **Bob issues a Capability Token (digital token) to Dave that:**
 - ✓ identifies Dave has the only subject entitled to use the token
 - ✓ states what Dave can perform (e.g. monitoring and configuring Bob's garden watering system)
 - ✓ states for how many days Dave can do these actions (i.e. token validity period)

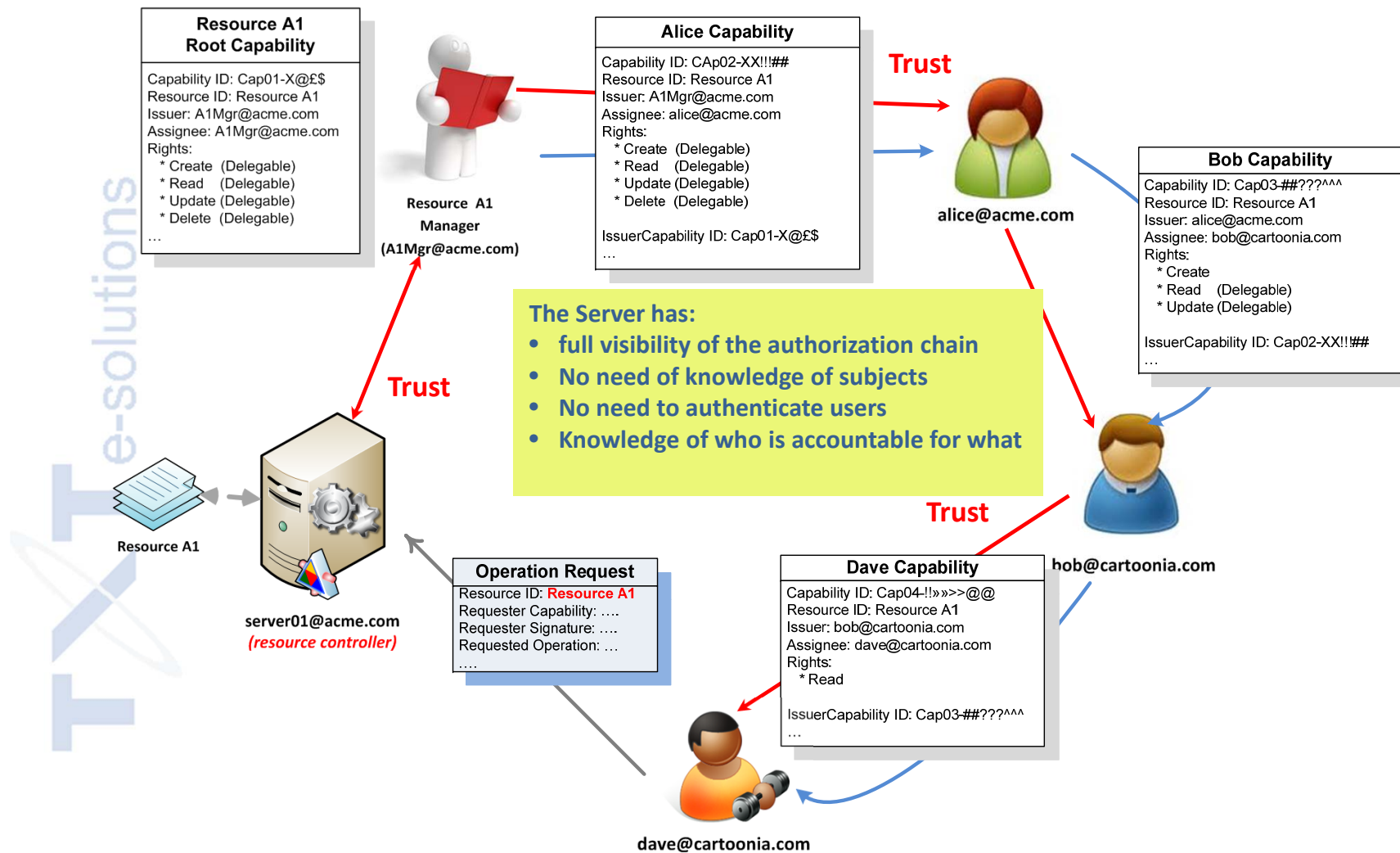


■ Pros:

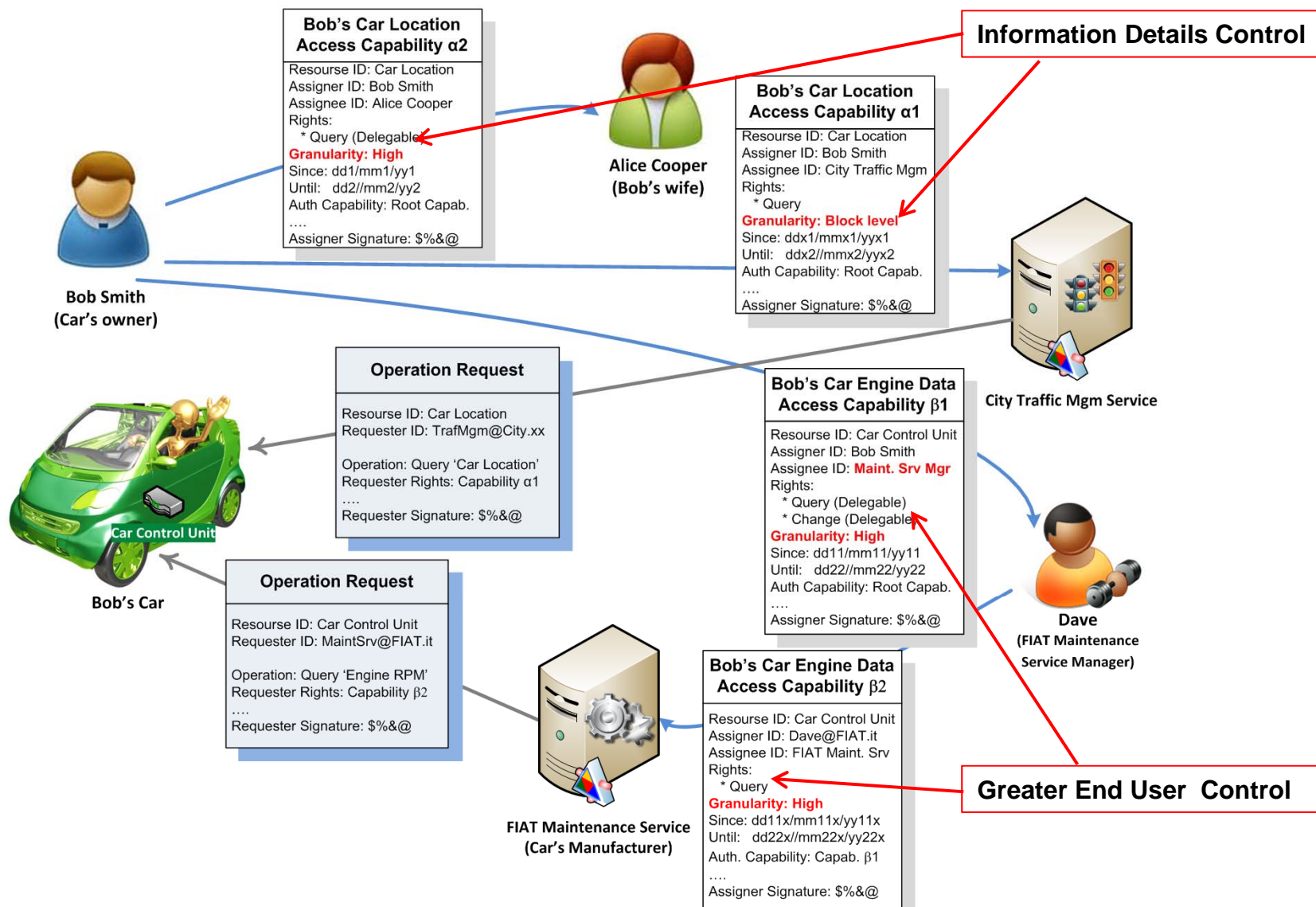
- ✓ Dave cannot use Bob's token for non-envisaged/non-authorized activities
- ✓ Dave cannot pass the token to someone else, nor can use it outside the validity period
- ✓ Easy to understand and easy to use approach (no rules to manage, fire&forget, ...)

- **Delegation Support:** a subject can grant the right to further delegate all or part of the granted rights to a third subject
- **Capability revocation:** capabilities can be revoked by properly authorized subjects
- **Information granularity:** a capability can grant access to a specific resource specifying also the level of detail granted
- **XML representation:** Capability Tokens have an XML representation (i.e. a digitally signed XML doc that can therefore be transferred by any means)
- **SAML/XACML based:** we use/extend SAML/XACML for capability token's elements

CapBAC Overall Scenario

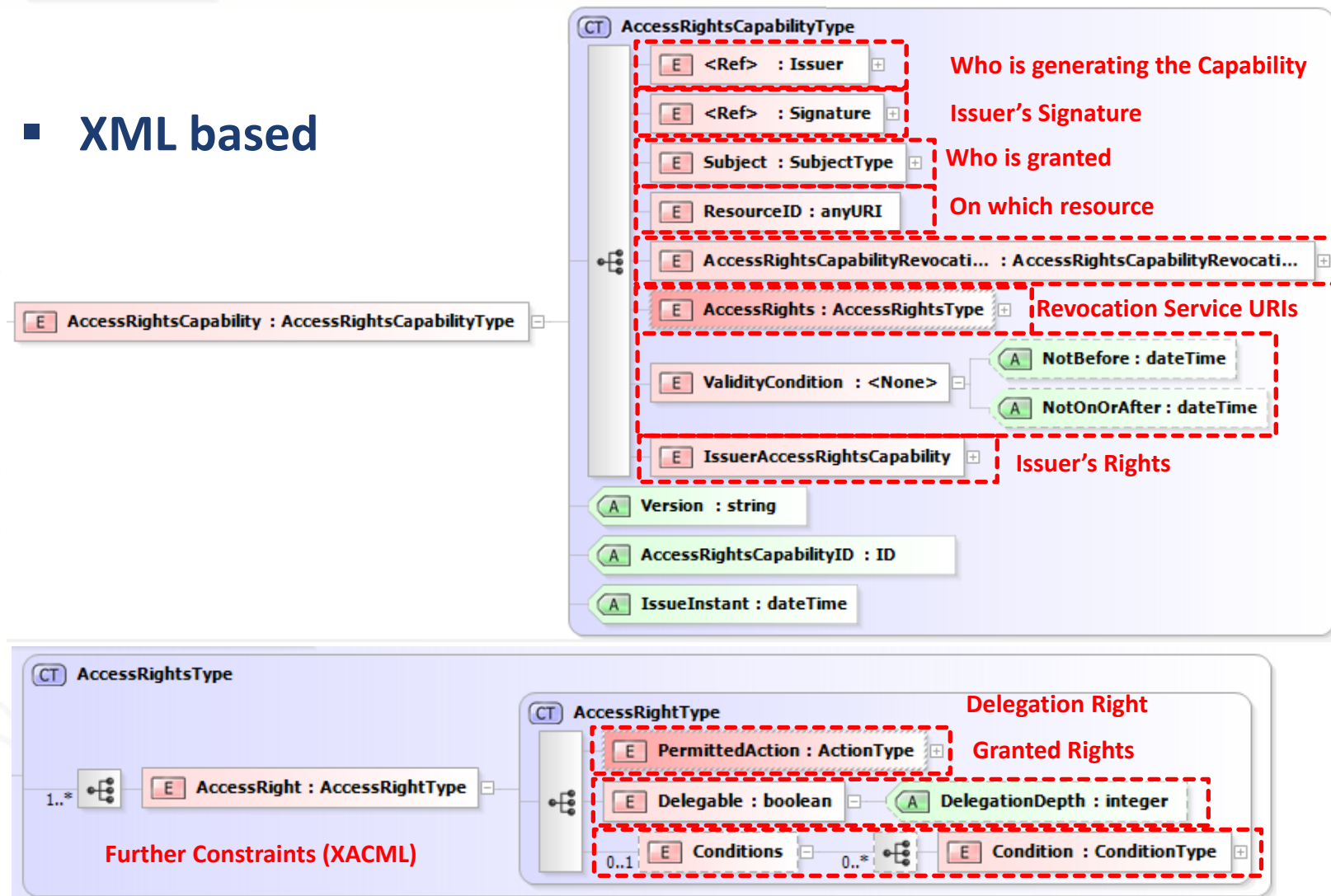


CapBAC Information Control



CapBAC Token Characteristics (1)

- XML based



CapBAC Token Characteristics (2)

■ Token Types:

➤ Root Capability Token: the 1st in chain Capability

- ✓ issued by the Owner of the identified resource
- ✓ Is a *normal* capability where:
 - ✓ the Granter and Grantee subjects are the same (**Issuer == Subject**)
 - ✓ The **IssuerAccessRightsCapability** (i.e. the *previous-in-chain* capability) is empty!

➤ Non Root Capability Token: a capability that has a predecessor

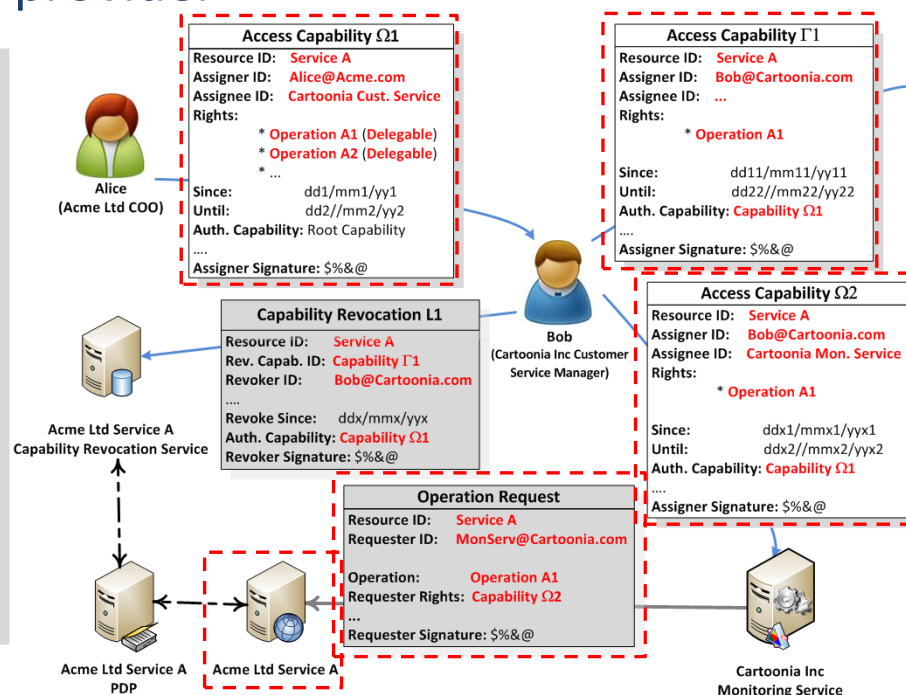
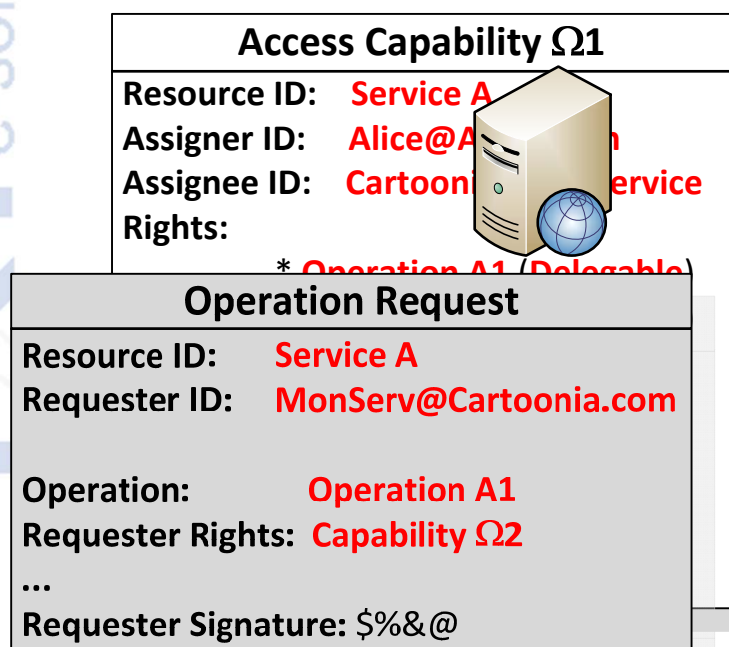
- ✓ The grants are a subset of its predecessor capability
- ✓ the Granter and Grantee subjects are normally different
- ✓ The **IssuerAccessRightsCapability** contains the predecessor capability

■ Chain of Capabilities:



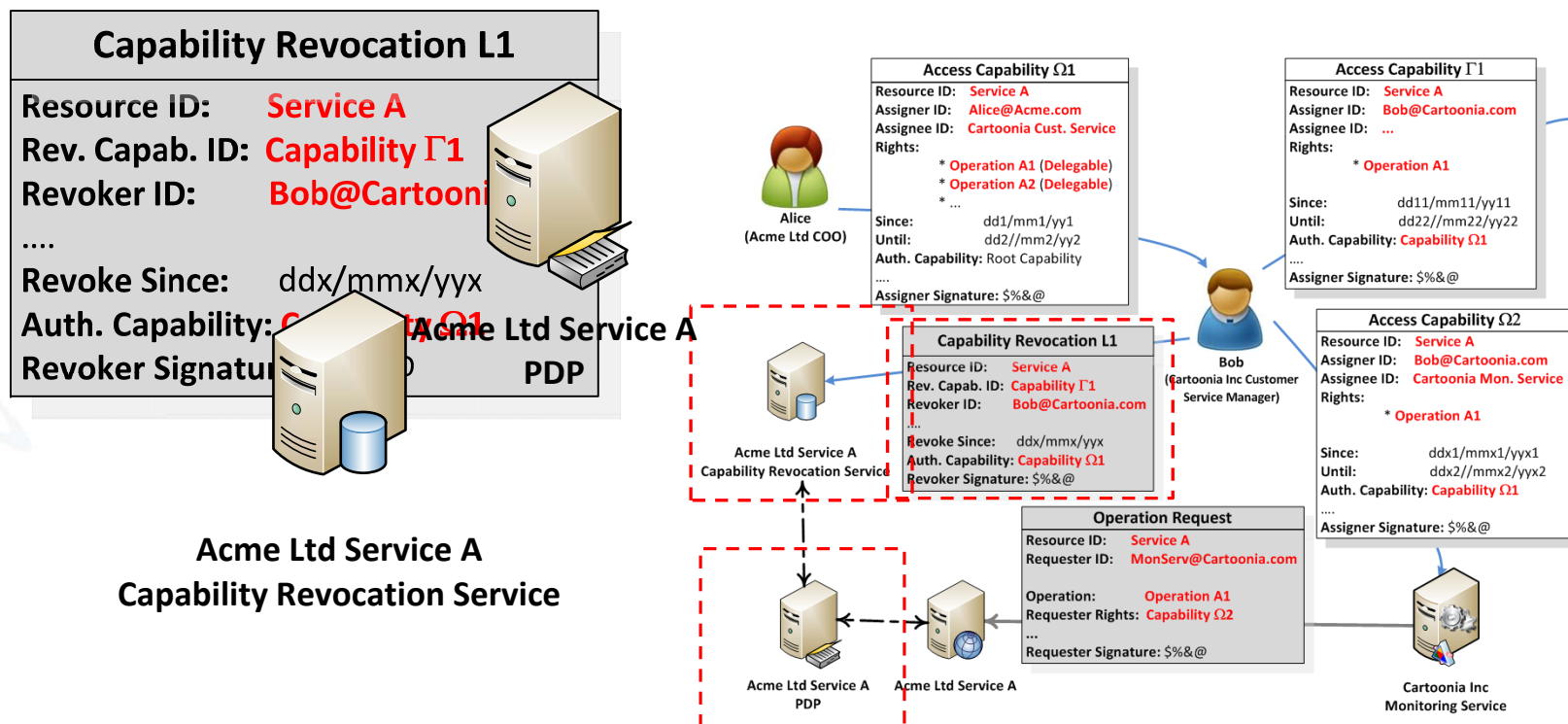
CapBAC Mandatory Functional Elements

- ❑ **Authorization capability:** details granted rights, delegation rights, the resource, the issuer, grantee
- ❑ **Resource manager:** service provider in charge of managing the identified resource
- ❑ **Service/operation request:** set of information to actually gain access to the resource via the service provider



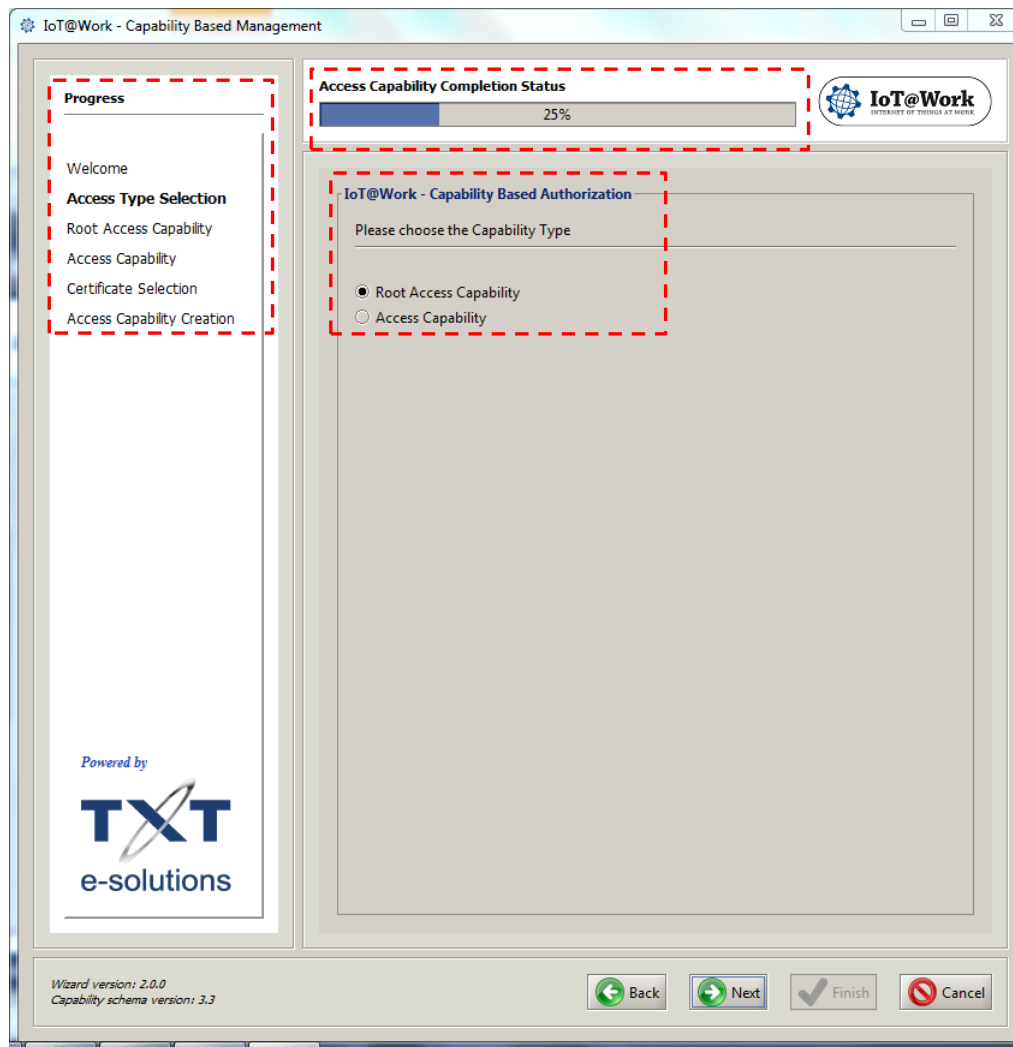
CapBAC Discretionary Functional Elements

- ❑ **Policy Decision Point:** service in charge of access request validation and decision
- ❑ **Capability Revocation Request:** revokes one or more capabilities
- ❑ **Capability Revocation Service:** in charge of managing capability revocations



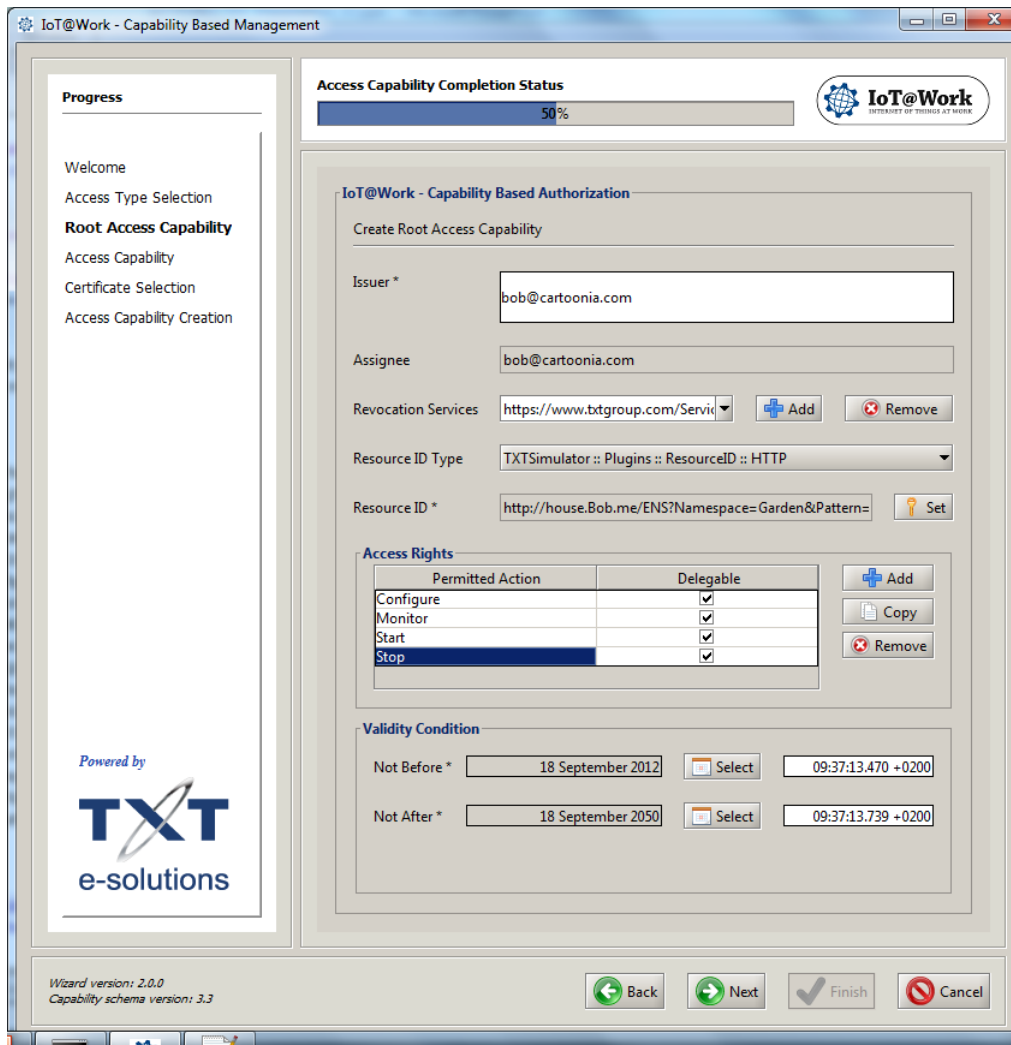
CapBAC Access Token Generation (1)

Currently using a stand alone Java-Based **Wizard** that guides the user (so anyone can run it)



CapBAC Access Token Generation (2)

Access token generation is not a **security sensitive** operation (so anyone can generate tokens)



The screenshot shows the 'IoT@Work - Capability Based Management' wizard. The 'Progress' pane on the left indicates the current step is 'Root Access Capability'. The main area is titled 'Access Capability Completion Status' with a 50% progress bar. Below this, the 'IoT@Work - Capability Based Authorization' section contains the following fields and controls:

- Create Root Access Capability**
- Issuer ***: bob@cartoonia.com
- Assignee**: bob@cartoonia.com
- Revocation Services**: https://www.txtgroup.com/Service (with + Add and - Remove buttons)
- Resource ID Type**: TXTSimulator :: Plugins :: ResourceID :: HTTP
- Resource ID ***: http://house.Bob.me/ENS?Namespace=Garden&Pattern= (with a Set button)
- Access Rights**: A table with 'Permitted Action' and 'Delegable' columns. The 'Stop' action is selected. Buttons for + Add, Copy, and - Remove are present.
- Validity Condition**: Fields for 'Not Before' (18 September 2012) and 'Not After' (18 September 2050), each with a calendar icon and a time offset (09:37:13.470 +0200).

At the bottom, the wizard version is 2.0.0 and the capability schema version is 3.3. Navigation buttons include Back, Next, Finish, and Cancel.

CapBAC Access Token Generation (3)



No big differences between *Root Tokens* and *Non Root* ones

IoT@Work - Capability Based Management

Progress

- Welcome
- Access Type Selection
- Root Access Capability**
- Access Capability
- Certificate Selection
- Access Capability Creation

Access Capability Completion Status 50%

IoT@Work - Capability Based Authorization

Create Root Access Capability

Issuer * bob@cartoonia.com

Assignee bob@cartoonia.com

Revocation Services https://www.bxtgroup.com/Service/ Add Remove

Resource ID Type TXTSimulator :: Plugins :: ResourceID :: HTTP

Resource ID * http://house.Bob.me/ENS?Namespace=Garden&Pattern= Set

Access Rights

Permitted Action	Delegable
Configure	<input checked="" type="checkbox"/>
Monitor	<input checked="" type="checkbox"/>
Start	<input checked="" type="checkbox"/>
Stop	<input checked="" type="checkbox"/>

+ Add Copy Remove

Validity Condition

Not Before * 18 September 2012 Select 09:37:13.470 +0200

Not After * 18 September 2050 Select 09:37:13.739 +0200

Powered by **TXT** e-solutions

Wizard version: 2.0.0
Capability schema version: 3.3

Back Next Finish Cancel

IoT@Work - Capability Based Management

Progress

- Welcome
- Access Type Selection
- Root Access Capability
- Access Capability**
- Certificate Selection
- Access Capability Creation

Access Capability Completion Status 50%

IoT@Work - Capability Based Authorization

Create Access Capability

Issuer bob@cartoonia.com

Assignee * dave@cartoonia.com

X509 Issuer Name

X509 Serial Number

Revocation Service https://www.bxtgroup.com/Services/Iot@Work/RevocationService...

Resource ID http://house.Bob.me/ENS?Namespace=Garden&Pattern=Mgm

Access Rights

Authorized Action Name	Authorize	Delegate
Configure	<input type="checkbox"/>	<input type="checkbox"/>
Monitor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Start	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Stop	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Validity Condition

Not Before * 23 September 2012 Select 09:57:58.412 +0200

Not After * 28 September 2012 Select 09:57:58.512 +0200

Powered by **TXT** e-solutions

Wizard version: 2.0.0
Capability schema version: 3.3

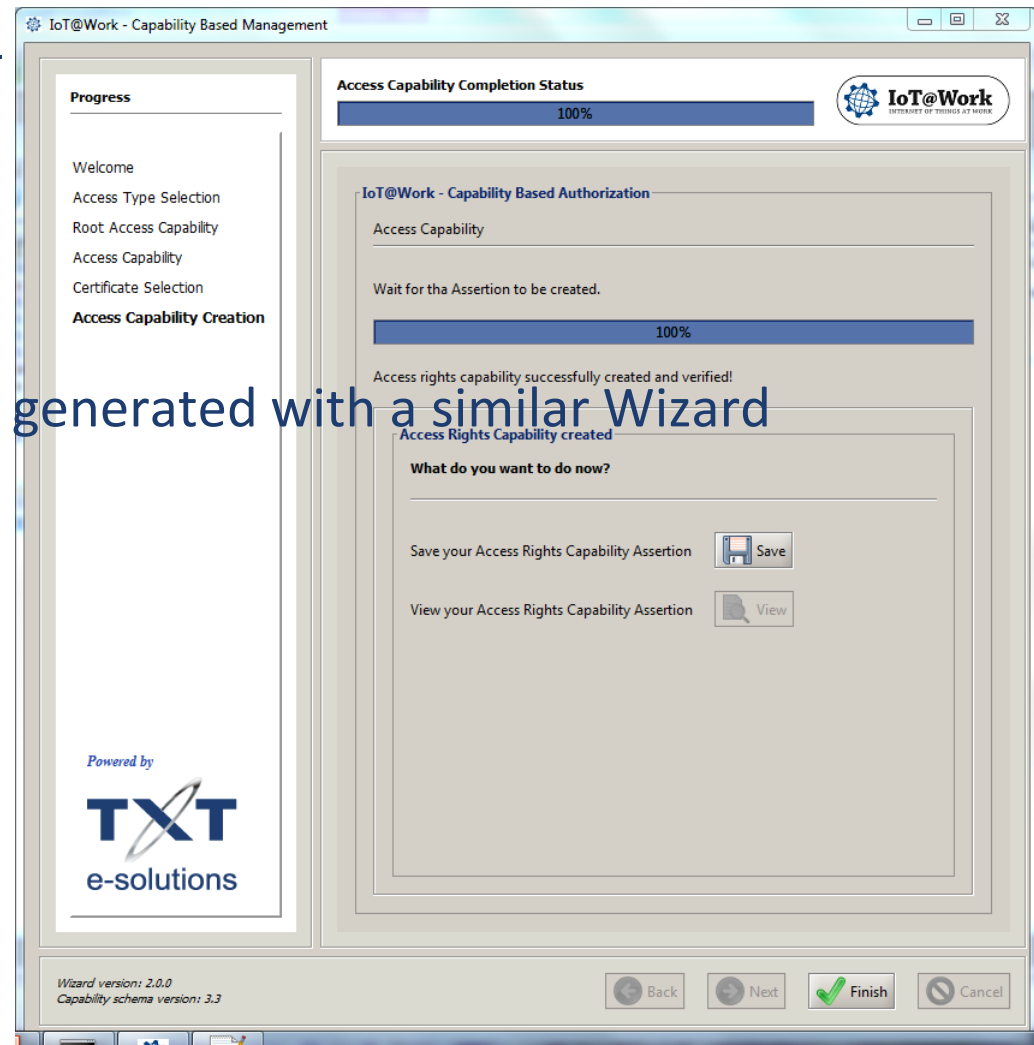
Back Next Finish Cancel

CapBAC Access Token Generation (4)

Tokens are **digitally signed XML docs**

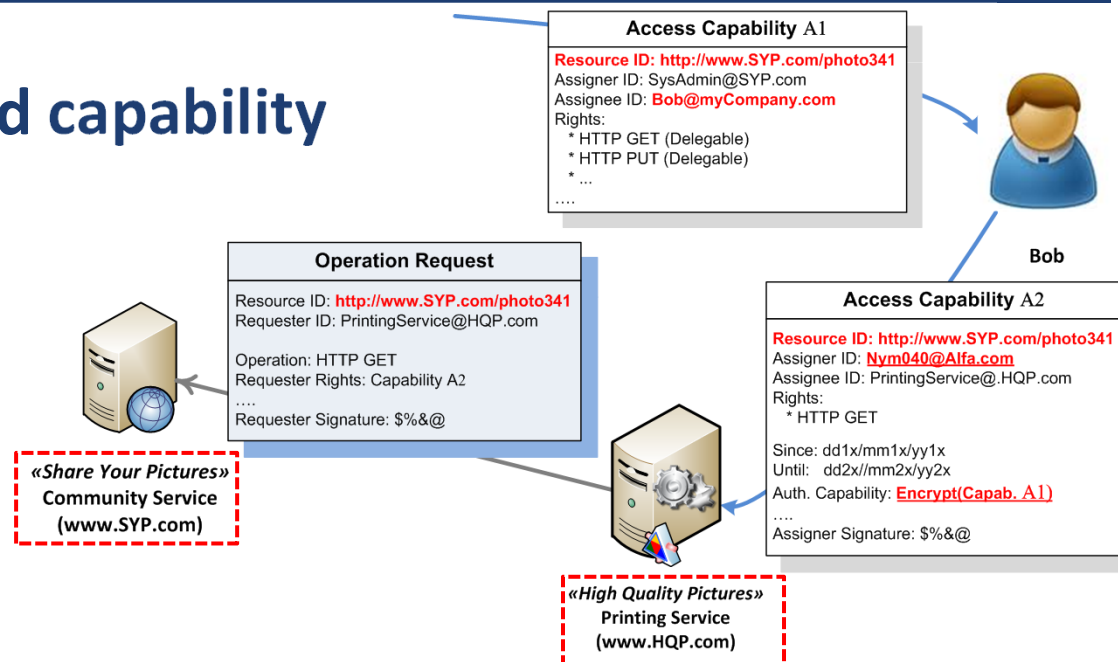
No real constraint to have a PKI

Revocation tokens are generated with a similar Wizard

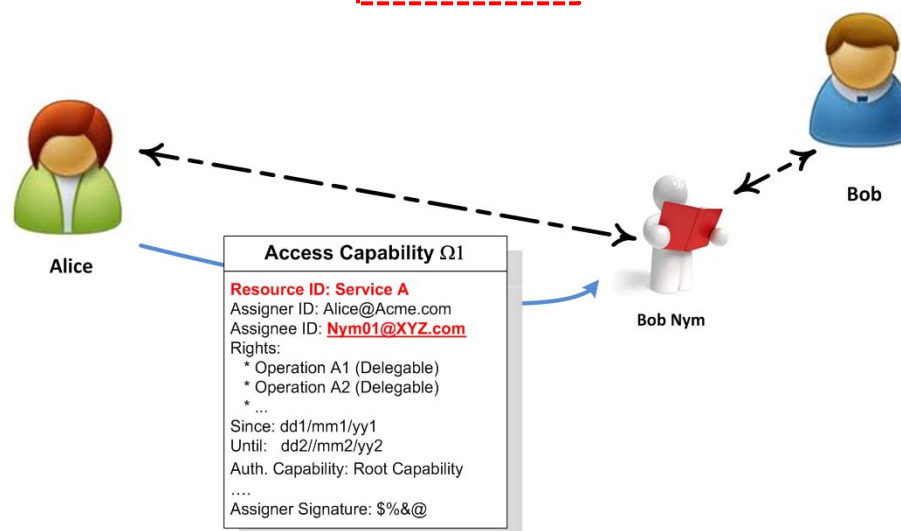


Privacy in Capability Based Authorization

■ Encrypted capability



■ Anonymous IDs



All development Java based

■ Tools/Libraries:

✓ **Wizards** (tools to be made available to all potentially involved subjects):

- Create Capability Tokens (or Capability Revocation Tokens)

✓ **Client/Server OSGi compliant library:**

- Manages capability selection on the client side
- Manages access request validation on the server side
- Implements IoT@Work ENS Authorization handshake (to be adapted for other services)

■ *CapBAC* services:

- ✓ **IoT@Work ENS Authorization Service:** can be used as a mock-up
- ✓ **CapBAC PDP Service:** checks requests against capability revocations
- ✓ **CapBAC Revocation Service:** manages capability revocation's requests
- ✓ **Client/server demo apps**

■ **Knobs tuning:**

- ✓ IoT@Work Capability provides many **Knobs**: **Validity period**, **Assigned rights**, **Capability revocation**, **Delegation**, **Delegation depth**, **Resource Granularity** , ...

■ **Capability Tokens:**

- ✓ XML verbosity
- ✓ EXI (Efficient XML Interchange) usage: CoAP, EXI \leftrightarrow XML interoperability
- ✓ More efficient encryption techniques (ECC, Id Based Encryption)
- ✓ issued capability tokens repository
- ✓ **Digital Ecosystem** approach to capability tokens generation for IoT (??)

■ **Usability:**

- ✓ Current wizards effectiveness
- ✓ Mobile devices usage

■ **Real Contexts Validation:**

- ✓ IoT@Work pilots
- ✓ FP7 IoT6
- ✓ Other contexts

Thanks for your attention!

Questions????