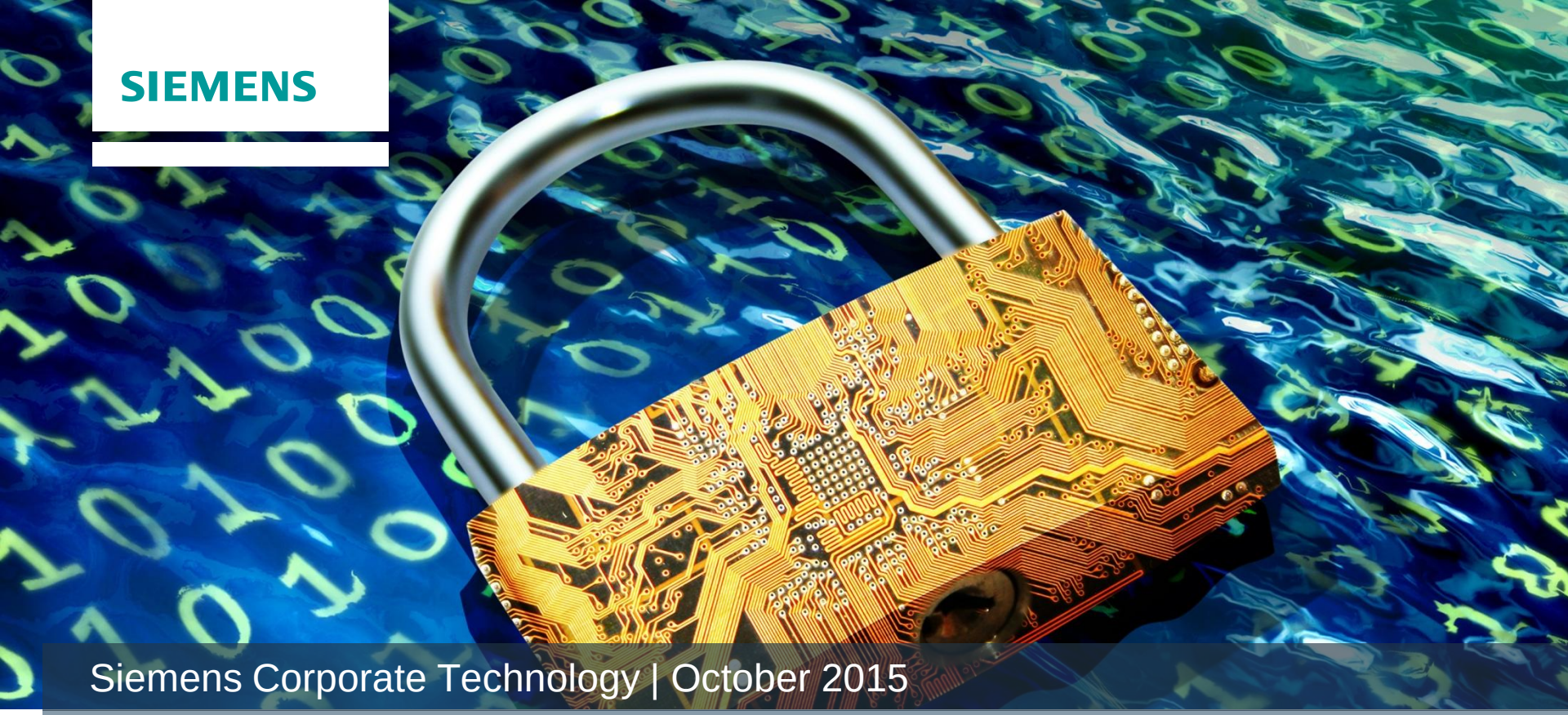
SIEMENS



Siemens Corporate Technology | October 2015

W3C WoT IG Meeting 29-30 October 2015, Sapporo

Landscape of Security & Privacy Means

Contents

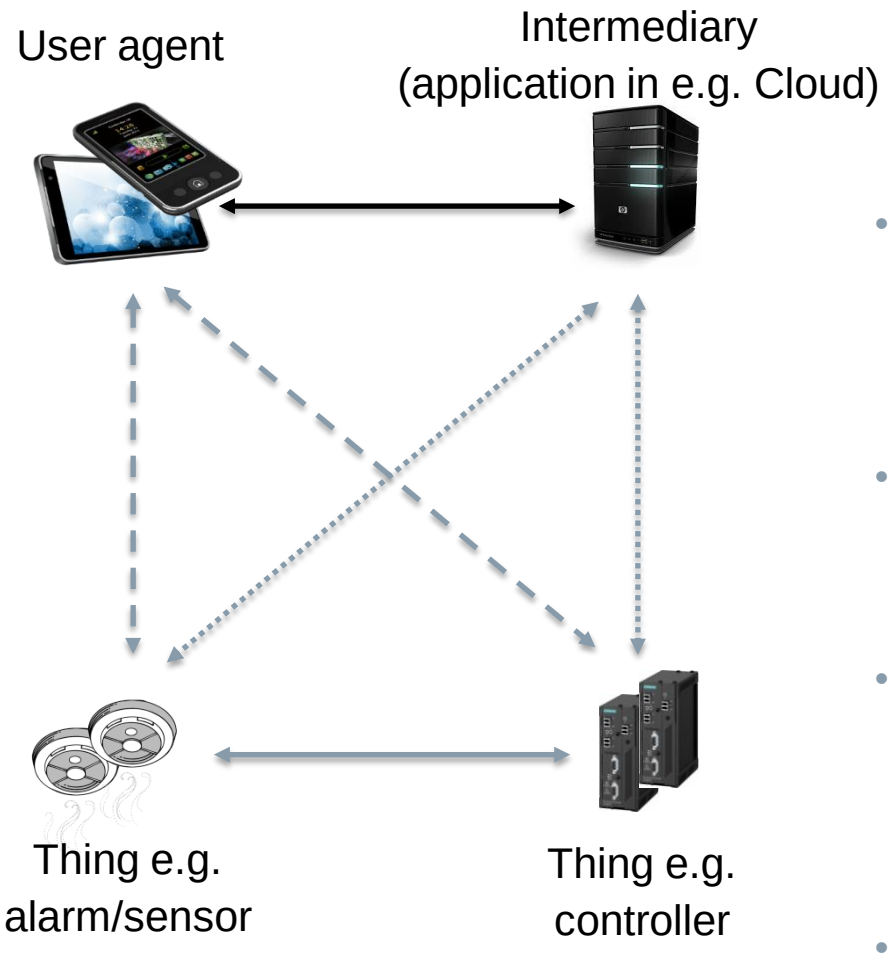
- Preface
- Setting-the-scene
- Things that matter
- State-of-the-art
- Conclusions

Preface

- This presentation digests the security and privacy landscape for WoT
- More details can be found in the WoT Wiki:
  - Overview page: Landscape of Security&Privacy Means
  - Details: Design-Time Security&Privacy Means

Setting-the-Scene

WoT Provides Distributed Systems



- Variety of components/actors e.g.
  - Things/devices
  - User agents (opt.)
  - Intermediaries (opt.)
- Variety of topologies e.g.
  - Direct interactions between (user agents and) things
  - Mediated interactions
- Variety of connectivity styles e.g.
  - Near field…wide-area
  - Intermittent…undisturbed
- Variety of communication patterns e.g.
  - Request/response
  - Publish/subscribe
  - One-way
- Variety of protocols e.g.
  - AMQP, CoAP, HTTP, MQTT, XMPP…

Setting-the-Scene
Protecting Distributed Systems

- The study of distributed systems protection started soon after their invention i.e. the 60/70ies
- So there is lots of prior art
- Corresponding work falls into following disciplines (examples for an online banking scenario):
  - **Privacy:** *understand/control the dissemination/use of personal information/resources*
    - *E.g. "John Doe happened to transfer 1000$"*
  - **Authorization:** *decide whether to execute an instruction*
    - *E.g. "transfer 1000$ from John Doe's bank account"*
  - **Authentication:** *establish confidence in claims that are made by system actors*
    - *E.g. "This is bank24.example" and "I am John Doe" or "I am over 18"*
  - **Secure communications and storage:** *assure that data is not eavesdropped or manipulated (e.g. 1000->10000) in exchange or storage*
  - **Provisioning and credentialing:** *establish/manage the underpinnings for most of above*
    - *E.g. registering John Doe as a user of the online banking and supplying John with means for transaction approval such as TANs*
- Key question: *what coverage does give the prior art for WoT?*

Characteristics/Dependencies of the Disciplines

- **Privacy**
  - Human-centric by definition
- **Authorization**
  - Builds upon authentication
  - Is very different for legal entity vs. individually-owned resources
- **Authentication**
  - Trusted-third parties preferred (in order to cope with functional/protocol complexity)
  - Leads to a differentiation into initial and subsequent authentication (SSO being a special case of subsequent authentication)
  - Builds upon provisioning (metadata) and credentialing
  - Establishes the prerequisite for authorization
- **Secure communications and storage**
  - Employ cryptographic mechanisms
  - Protocol stack layer allocation matters
  - Builds upon credentialing
- **Provisioning and credentialing**
  - Establishes prerequisites for authentication, secure communications and storage

Aspects of these Disciplines

- **Privacy**
  - Informational self-determination
  - Anonymization, pseudonymization
- **Authorization**
  - Authorization management
  - Authorization enforcement
- **Authentication**
  - Initial authentication
  - Subsequent authentication, SSO
- **Secure communications and storage**
  - Data origin authentication
  - Confidentiality
- **Provisioning and credentialing**
  - Provisioning
  - Credentialing

Means to Address the Security Goals

SIEMENS

- **Cryptographic primitives**: algorithms to transform data
  - Encryption vs. message authentication
  - Asymmetric (e.g. RSA, ECDSA) vs. symmetric (e.g. AES, SHA-2)
- **Cryptographic objects**: representations of transformed data along with metadata e.g. JOSE
  - Form factors: ASN.1 (e.g. PKCS), XML (e.g. XML Signature/Encryption), JSON (e.g. JOSE), CBOR (e.g. COSE)...
- **Security tokens**: formats to make assessments about system actors e.g. JWT
  - Form factors: ASN.1 (e.g. Kerberos tickets), XML (e.g. SAML assertions), JSON/CBOR (e.g. JWT)…
- **Security protocols**: means to exchange cryptographic objects or security tokens
  - Focus on exchanging cryptographically transformed data:
    - Transport-bound (protection of data-in-transit): SSL/TLS, DTLS, DICE
    - Information-bound (protection of data-at-rest): CMS, XML Signature/Encryption, JOSE/COSE — potentially enhanced by IoT/WoT-adequate freshness indicators
  - Focus on requesting/submitting security tokens:
    - Kerberos protocol (Kerberos ticket), SAML protocol (SAML assertions), OAuth (OAuth tokens, note: contents not defined), OIDC (JWT)

Setting-the-Scene
WoT Has Specific Needs

- **Inclusion of physical goods**
  - Requires to reflect use cases such as change-of-ownership (tends to be ignored/naively solved in digital goods-only systems)
- **Constrained devices**
  - Callers resp. callees may have (severe) limitations on power supply. processing power, volatile/static memory, lack of IO devices/user attention, software/configuration update…
- **Constrained networks**
  - Callers resp. callees interact across low bandwidth, lossy networks possibly with intermittent communications…
- **Not only human users**
  - Number of callers that are to-be-authenticated grows by a factor of ca. 10
- **Not only IT-applications**
  - Number of callees that are to-be-authenticated: grows by a factor of ca. 10000
- **Connectivity, de-perimeterization**
  - User agents connect from public networks increasing the attack surface (not WoT-specific)
- More info: Security&Privacy Challenges

Digital vs. Physical Goods

- The IT industry is used to processing digital goods. It has a ‘digital good’ mindset:
  - **Digital goods** — *reproduction, relocation of item instances at almost no cost*
  - Examples: Web pages, messages, contact/mapping information, mp3 files…
  - Aspects:
    - Static vs. dynamic objects
    - Human vs. machine-readable
- Things are physical:
  - **Physical goods** — *reproduction, relocation of item instances at cost*
  - Examples: lighting devices, smoke sensors, thermostats, controllers…
  - Aspects:
    - Consumer vs. investment goods
    - Individually vs. legal entity-owned

Technology Generations

- **Classic**
  - Invented <2010
  - Native to enterprise/office IT and the traditional Web - *possibly no or only a partial fit for WoT/IoT*
  - Examples: Kerberos, LDAP, P3P, PKIX, S/MIME, SAML, SSL/TLS
- **New**
  - Invented 2010-2015
  - Addressing new Web application styles (Web/REST APIs, mobile/browser-based apps), not native to WoT/IoT - possibly no or only a partial fit for WoT/IoT
  - Examples: FIDO, JOSE, OAuth, OIDC, SCIM
- **Future**
  - Invented >2015
  - Native to WoT/IoT
  - Examples: ACE (not an individual mechanism but a container for many mechanisms), COSE, DICE

Interoperability

- It makes a difference if a WoT security and privacy solution can be a silo or has to be interoperable:
  - Silo'ed solutions are able to support **same-domain scenarios** where e.g. a single vendor/provider supplies all components that interact in a distributed system. Standards for security and privacy are optional. They facilitate reuse
  - Interoperable solutions are required to support **cross-domain scenarios** where there is no such single vendor. Standards for security and privacy are mandatory. They provide interoperability AND reuse.
- Hypothesis: current IoT/WoT projects either neglect security and privacy or create a silo'ed solution.

Silo'ed vs. Interoperable for Traditional Web (1)

- **Privacy**: DIY (ubiquitous) or P3P (some)
- **Authorization**: DIY
  - There are authorization solutions but they are silo'ed since there is no standard that is commonly accepted
- **Authentication**:
  - Server authentication: SSL/TLS (ubiquitous)
  - User or client authentication:
    - Initial: DIY (ubiquitous) or HTTP Basic/Digest (some)
    - Subsequent: DIY ("SSO cookies", ubiquitous) or SAML/WS-Fed/OpenID/OIDC (some)
- **Secure communications and storage**
  - Transport-bound: SSL/TLS (ubiquitous)
  - Information-bound: PKCS#7/CMS, XML Signature/Encryption (some)
- **Provisioning and credentialing**: DIY (ubiquitous) or CMP/KeyProv/PKCS (some)

Silo'ed vs. Interoperable for Traditional Web (2)

- Filter the security and privacy practices in the traditional Web according the keywords "*standard*" and "*ubiquitous*" then there only is a single mechanism: **SSL/TLS**
  - It delivers **secure communications** (transport-bound) and **server authentication**
  - It does not deliver privacy (beyond encrypting data in transit), authorization, user authentication, provisioning and credentialing
- But there are lots of security mechanisms in Web-based applications esp. for authorization and user authentication - *of course your bank does implement authorization features on your bank account*
- So most security functionality is DIY
- Key question: *is DIY security and privacy viable for WoT?*
  - Seems viable for the traditional Web but already encounters issues there (when it comes to use cases where #n>1 service provides want to deliver functionality to the same users)
  - Started to be unviable with new Web application styles:
    - The famous use case just not covered by 'classic' technology items is: ***I** want **office24.com** to print **my photos** stored at **Google Drive***
    - This led to the rogue wave of security mechanism innovation around OAuth
  - Believed to be unviable for WoT too – *okay that's a claim. So let's see hands…*

Impact

- Some security and privacy technologies affect the **product core** from a solution design and realization perspective e.g. authentication and authorization including the representation and handling of authenticated callers throughout a codebase – ***everybody*** *in WoT R&D teams should have an understanding*
- Other security technologies may be **added** or **replaced** easily such as transport-oriented security mechanisms – ***few*** *in WoT R&D teams need to understand them*

State-Of-The-Art
Privacy

- **Classic**:
  - Focus: express privacy practices and match with user expectation
  - Corresponding building blocks: P3P
  - Note: authorization and secure communication mechanisms can serve as privacy-enhancing mechanisms. They are covered in other sections and not considered here
- **New**:
  - Focus: (self-)manage access to user-specific information and resources
  - Corresponding building blocks: UMA
  - Note: UMA actually is an authorization technology that inherits from OAuth
- **Future**:
  - Objective: express privacy practices and match with user expectation, (self-)manage access to user-specific information and resources
  - Initiatives: ACE (PET, UMA-for-CoAP)

State-Of-The-Art
Authorization

- **Classic**:
  - Focus: express and enforce authorization for legal entity-owned resources
  - Corresponding building blocks: RBAC, XACML
- **New**:
  - Focus: express and enforce authorization for individually-owned resources
  - Corresponding building blocks: OAuth
- **Future**:
  - Objective: express and enforce authorization for legal entity as well as individually-owned resources, downscale/optimize processing and representation demands for that purpose
  - Initiatives: ACE (CoRE Authz, DCAF, OAuth-for-CoAP)

State-Of-The-Art
Authentication

- **Classic**:
  - Focus: conducting initial user authentication with shared secrets, shared secret keys, public/private keys, requesting/reporting on initial user authentication
  - Corresponding building blocks:
    - Initial authentication: EAP, LDAP, OATH, SSL/TLS
    - Subsequent authentication: Kerberos, SAML, WS-Federation
- **New**:
  - Focus: ditto plus conducting initial client authentication with shared secrets requesting/reporting on initial client authentication
  - Corresponding building blocks:
    - Initial authentication: FIDO
    - Subsequent authentication: JWT, OAuth (client authentication), OIDC
- **Future**:
  - Objective: authenticate things and request/report on things authentication, downscale/optimize processing and representation demands for that purpose
  - Initiatives: ACE (TWAI)

State-Of-The-Art
Secure Communications and Storage

- **Classic**:
  - Focus: cryptographic transformation of data in transport, expressing cryptographically transformed (application) data
  - Corresponding building blocks:
    - Transport-bound: IPSec, SSH, SSL/TLS and DTLS
    - Information-bound: CMS/PKCS, PGP, S/MIME, XML Signature and XML Encryption
- **New**:
  - Focus: ditto
  - Corresponding building blocks:
    - Transport-bound: DICE
    - Information-bound: JOSE
- **Future**:
  - Objective: downscale/optimize processing and representation demands
  - Initiatives:
    - ACE (OSCOAP)
    - COSE

State-Of-The-Art
Provisioning and Credentialing

- **Classic**:
  - Focus: describing the digital identify (identifiers, attributes, affiliations) of human users and supply them with credentials
  - Corresponding building blocks: PKIX, EKMI, KeyProv, SPML
- **New**:
  - Focus: ditto
  - Corresponding building blocks: SCIM
- **Future**:
  - Objective: describe things and supply them with credentials, downscale/optimize processing and representation demands for that purpose
  - Initiatives:
    - W3C WoT [TF-TD] addresses things description, not things identity, credentialing
    - Kantara IdoT addresses things identity
    - IETF ACE addressed a method to supply keys for devices node when they initially join (expired draft)

Conclusions
Maturity, Usage, WoT Fitness

- **Classic**
  - Maturity: **very high** esp. Kerberos, LDAP, P3P, PKIX, S/MIME, SAML, SSL/TLS
  - Usage: **large-scale production** use esp. Kerberos, LDAP, S/MIME, SSL/TLS
  - WoT fitness: **little to none**
- **New**
  - Maturity: **high** esp. JOSE, OAuth, OIDC, SCIM
  - Usage: **large-scale production** use esp. OAuth, OIDC
  - WoT fitness: **limited**
- **Future**
  - Maturity: **low** (as of now, not meant to be any surprise)
  - Usage: **experimental to none**
  - WoT fitness: **high**

Conclusions
Implication

- Observation:
  - If one is interested in **interoperable** security and privacy solutions for WoT, then one **can not just go ahead**: work is still to be done
  - If one is only interested in a **silo'ed** security and privacy solution for WoT, then one can **go ahead** and pick/build from the already existing offerings
- Implication:
  - Security and privacy approaches that are currently found in WoT/IoT projects are silo'ed (which is no problem when they are meant and advertised as silo'ed)

Conclusions
Fitness From Another Angle

- The well-known Internet/Web security mechanisms do **not match** class 1/0 devices
- Results in a need to optimize/innovate security mechanisms
- Optimization needs include:
  - **Down-scaling** of security system implementations
  - **Lightweight security** mechanisms covering
    - Cryptographic primitives
    - Cryptographic objects
    - Security tokens
    - Security protocols

Conclusions
Whitespots

- Besides adaptations of existing security mechanisms, new or extended security mechanisms and functionality are needed to address the whitespaces:
  - Managing the authorization to authorize
    - Small print: not an overall white-spot (IETF ACE has some work) but often overseen and hence mentioned here
  - Things discovery authorization
  - APIs resp. software structure with respect to security and privacy functionality
  - …

Wrap-Up

- **Classic and new security and privacy technologies (only) not do the trick for WoT**
  - Different constraints: adaptations and optimizations needed
  - White-spots: innovation needed
- Such technologies are in their **incubation** (called “future” for this reason). IETF ACE takes a **leading position** in their development:
  - Suggesting a (new) **trusted 4th party** supporting/representing the requesting party domain (classical/new approaches consider trusted 3rd parties supporting/representing the resource owner domain)
  - This new component currently focuses on the authentication and authorization enabling but might also offer help with respect to provisioning and credentialing (not yet explored)
  - Focuses on the bits exchanged in the network rather than e.g. APIs or software structure
  - Caveats:
    - Shake-out is needed: lots of ideas and starting points, many at brainstorming stage (*yellow vs. green bananas*)
    - After shake-out of the protocol building blocks, things can still be screwed up in implementation (e.g. accidental implementation/deployment errors). Additional means such as best practice recommendations, compliance/sanity/penetration tests are needed in addition

Author

Oliver Pfaff, Siemens AG, CT RTC ITS