AUTHENTICATION II

Introduction to Computer and Network Security

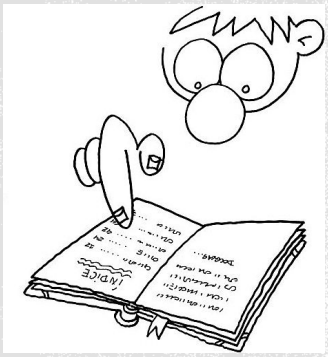
Silvio Ranise [silvio.ranise@unitn.it or *ranise@fbk.eu*]

UNIVERSITÀ DI TRENTO

FONDAZIONE BRUNO KESSLER

CONTENTS

- Single Sign On
- Security Assertion Markup Language (SAML)
  - Use case
  - Overview
  - Some details
  - Security considerations
- Identity Infrastructures
  - National: SPID & CIE
  - European: eIDAS



2 SINGLE SIGN ON (SSO)

Outsourcing authentication to trusted 3rd party identity providers

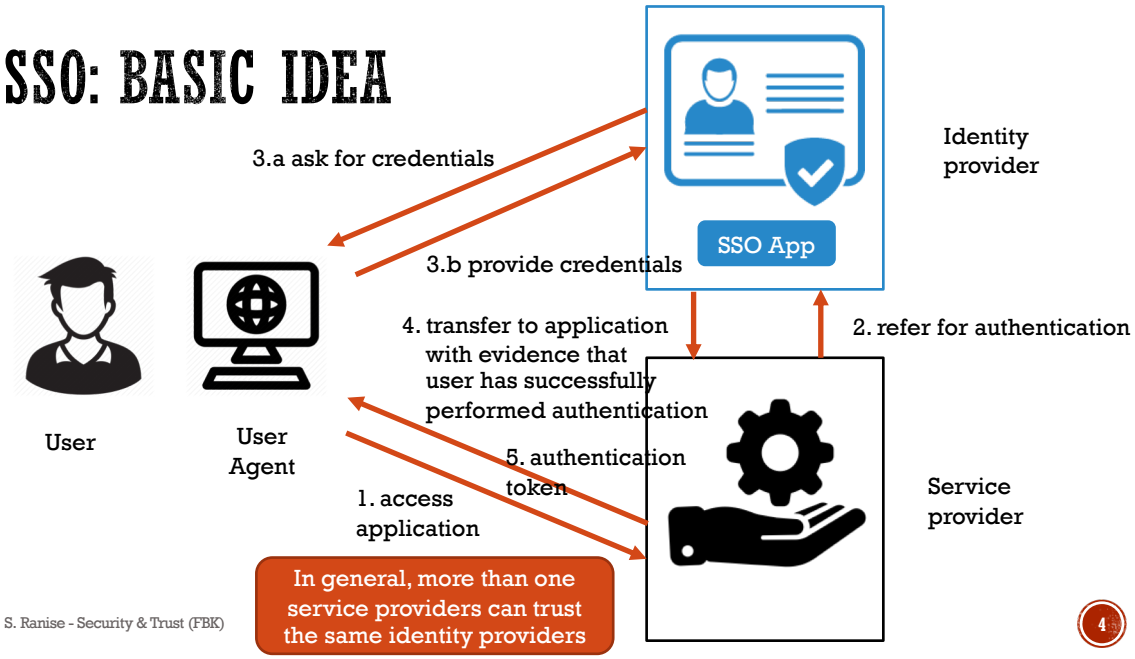
WHY SSO IS NEEDED?

- Multiple systems typically require multiple sign-on dialogues
  - Password fatigue
    - **Multiple sets of credentials**
    - **Presenting credentials multiple times**
- Headache for administration and users
- The more security domains, the more sign-ons required

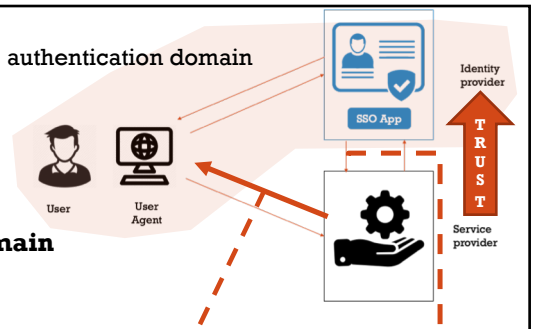
Security domain: an application or collection of applications trusting a common security token for authentication, authorization or session management

A **security token** is issued to users after they have actively authenticated with their identifiers and credentials (e.g., passwords or other authentication factors) to the security domain

SSO: BASIC IDEA



SSO PROPERTIES



- **Credentials never leave the authentication domain**
  - I.e. identity provider + user agent
- **Service providers** (affiliated domains) **have to trust the authentication domain**
  - Credentials must be asserted correctly
  - Protect from unauthorised use
- **Authentication transfer has to be protected**
  - Replay prevention on authentication tokens
    - Need to provide nonce as a freshness parameter to limit the **reuse** and **lifetime** of the authentication tokens/assertions
  - Interception/masquerade attacks
    - Need to add cryptographic signatures to authentication assertions

User is entitled to access the provided services

Authentication token

User has been authenticated by means of a password

Authentication assertion

SAML: INTRODUCTION

Security service for authentication that can be configured to fit the goals of several different use case scenarios

SAML HISTORY AND KEY FEATURES

- SAML = Security Assertion Markup Language
- SAML 1.0 defined in **2002**, SAML 2.0 defined in **2005**
- A common **language** and **flow** between systems that want to provide an SSO experience to users
- Answer to the lack of standards and interoperable solutions for exchanging authentication and authorization information across security domains
  - **Heavily based on mechanisms implemented in browsers** (e.g., redirections)

SAML OVERVIEW

- SAML = Security Assertion Markup Language
- Data format for exchanging
  - Authentication data
  - Authorization data
- XML-based
  - XML = Extensible Markup Language
  - https://www.w3.org/TR/xml11/
- An open standard from OASIS
  - OASIS = Organization for the Advancement of Structured Information Standards

- XML is a software- and hardware-independent framework for storing and transporting data
- XML is a markup language much like HTML
  - XML was designed to carry data with focus on what data is
  - HTML was designed to display data - with focus on how data looks
  - XML tags are not predefined like HTML tags are
- XML was designed to **store** and **transport** data
- XML was designed to be self-descriptive (whatever this means)

SAML FEDERATION

- SAML distinguish two main entities
  - Identity Provider (IdP)
  - Service Provider (SP)
- Federation is a group of entities
  - Sharing a common policy
  - Managed as a single entity

Identity Provider (IdP)
- Authenticates the user
- Provides authorisation information

Trust relation

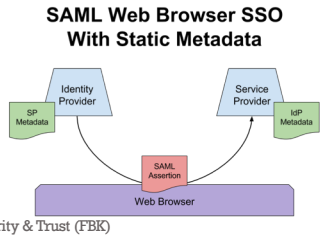
Trust establishment is performed by exchanging so called **metadata**

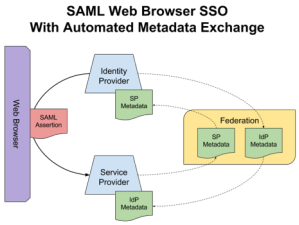
Service Provider (SP)
- A server that hosts protected resources
- It relies on information provided by the IdP
- Local access policies to regulate access to protected resources

- The authentication is always between a User and an IdP
- The federation establish only the **initial trust** among the resources

ON METADATA AND TRUST ESTABLISHMENT

- IdP and SPs share metadata in whatever form and by whatever means possible
- At least the following metadata must be shared:
  - Entity ID (globally-unique identifier included in every message issued by the entity)
  - Cryptographic keys
    - For authentication purposes, a SAML message may be digitally signed by the issuer and content of message can be protected by a public encryption key belonging to the ultimate receiver. Indeed, trusted public keys must be shared in advance.



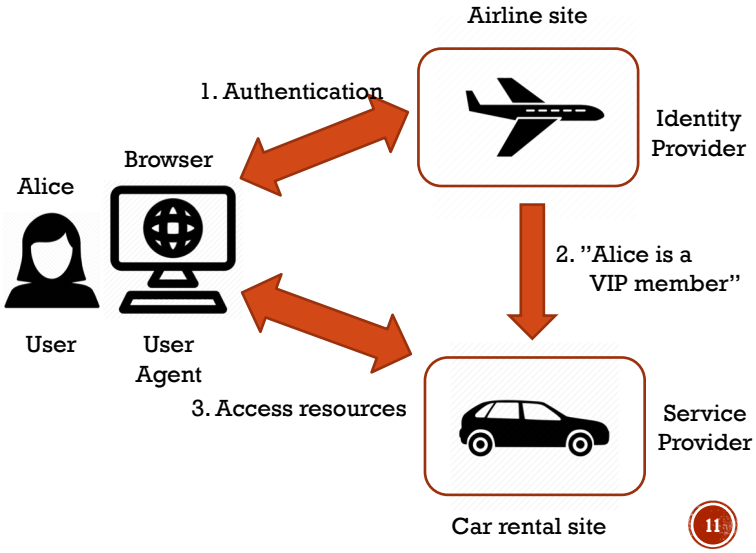


More details can be found at https://en.wikipedia.org/wiki/SAML_metadata

SAML: A POSSIBLE SCENARIO (1)

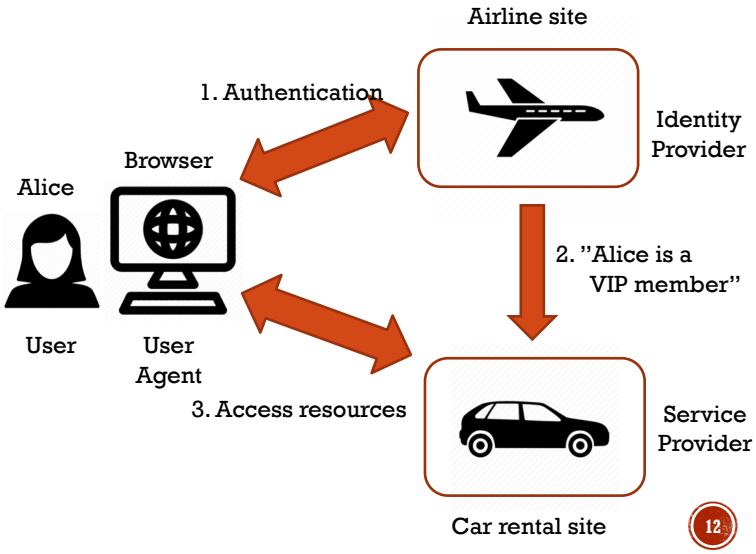
Note: to be precise, the Airline site plays two roles: SP and IdP

- Consider Alice visits an airline website for making her trip
- For booking her flight, she provides her credentials to airline website
- After booking, she found a link to car rental (from airline website)
- She visits car rental website



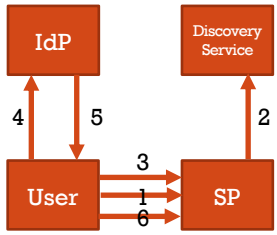
SAML: A POSSIBLE SCENARIO (2)

- Alice rents a car without signing in again…
- … because the car rental site trusts the airline site when transmitting authentication authentication assertions such as 2



SAML AUTHENTICATION FLOW: OVERVIEW

1. A user want to access an SP
2. The user is redirected to a Discovery Service
   - It can be an external service or embedded in the SP
   - Allow the user to choose the IdP
3. The user goes back to the SP with the ID of his/her own IdP
4. The user is redirected to the IdP
5. Authentication is performed
6. The user goes back to the SP with the authentication



All the steps above are associated with a SAML assertion (in XML format)

SAML AUTHENTICATION FLOW: REMARKS

- **SAML Authentication was mainly designed for Web Services**
- It is possible to support other type of resources
  - Either web based or native applications
- Resources can support multiple SAML profiles
  - The **profile** identifies the exchange **protocol** and the **message format**
- Most widely used profile for web applications is **redirect**
  - The browser shows a page with a Javascript performing the redirect

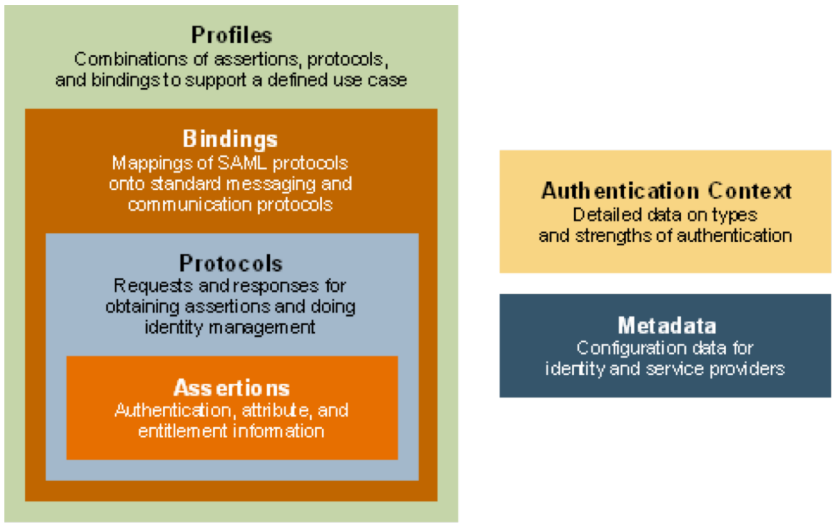
15

SAML: SOME DETAILS

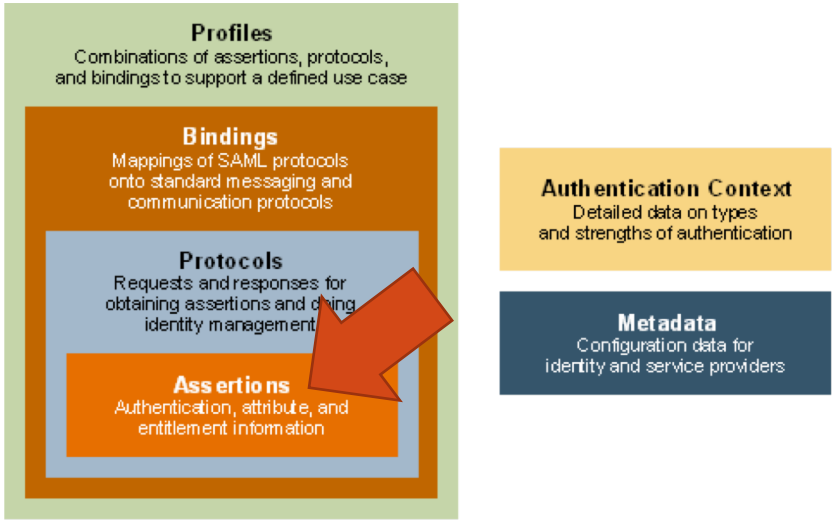
More information available at

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

SAML OVERVIEW



SAML OVERVIEW



ASSERTIONS AND ITS TYPES

- **Assertion** = set of statements (claims) made by a SAML authority (asserting party)
  - It can be seen as the **unit of information exchanged in SAML**
- **Authentication assertion**
  - Issued by a party that authenticates users
  - It describes
    - Who issued the assertion
    - Authenticated Subject
    - Validity period
    - Other authentication-related information
- **Attribute assertion**
  - It defines specific details about the Subject
  - *Example*: 'Alice' has 'VIP member' status
- **Authorization assertion**
  - It defines something the Subject is entitled to do
  - *Example*: 'Alice' is permitted to rent a car when on a business trip

SAML AUTHN ASSERTION: EXAMPLE

```
 1: <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 2:   Version="2.0"
 3:   IssueInstant="2005-01-31T12:00:00Z">
 4:   <saml:Issuer Format=urn:oasis:names:SAML:2.0:nameid-format:entity>
 5:     http://idp.example.org
 6:   </saml:Issuer>
 7:   <saml:Subject>
 8:     <saml:NameID
 9:       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
10:         j.doe@example.com
11:     </saml:NameID>
12:   </saml:Subject>
13:   <saml:Conditions
14:     NotBefore="2005-01-31T12:00:00Z"
15:     NotOnOrAfter="2005-01-31T12:10:00Z">
16:   </saml:Conditions>
17:   <saml:AuthnStatement
18:     AuthnInstant="2005-01-31T12:00:00Z" SessionIndex="67775277772">
19:     <saml:AuthnContext>
20:       <saml:AuthnContextClassRef>
21:         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
22:       </saml:AuthnContextClassRef>
23:     </saml:AuthnContext>
24:   </saml:AuthnStatement>
25: </saml:Assertion>
```

SAML AUTHN ASSERTION: EXAMPLE

```
 1: <saml:Assertion xmlns:saml="urn:oasis:na                    n"
 2:   Version="2.0"
 3:   IssueInstant="2005-01-31T12:00:00Z">
 4:   <saml:Issuer Format=urn:oasis:names:SAML:2.0:nameid-format:entity>
 5:     http://idp.example.org
 6:   </saml:Issuer>
 7:   <saml:Subject>
 8:     <saml:NameID
 9:       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
10:         j.doe@example.com
11:     </saml:NameID>
12:   </saml:Subject>
13:   <saml:Conditions
14:     NotBefore="2005-01-31T12:0
15:     NotOnOrAfter="2005-01-31T1
16:   </saml:Conditions>
17:   <saml:AuthnStatement
18:     AuthnInstant="2005-01-31T12:00:00Z" SessionIndex="67775277772">
19:     <saml:AuthnContext>
20:       <saml:AuthnContextClassRef>
21:         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
22:       </saml:AuthnContextClassRef>
23:     </saml:AuthnContext>
24:   </saml:AuthnStatement>
25: </saml:Assertion>
```

SAML authority making the claim

entity about which the claim is being made

SAML AUTHN ASSERTION: EXAMPLE

```
 1: <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 2:   Version="2.0"
 3:   IssueInstant="2005-01-31T12:00:00Z">
 4:   <saml:Issuer Format=urn:oasis:names:SAML:2.0:nameid-format:entity>
 5:     http://idp.example.org
 6:   </saml:Issuer>
 7:   <saml:Subject>
 8:     <saml:NameID
 9:       Format="urn:oasis:names:tc                    emailAddress">
10:         j.doe@example.com
11:     </saml:NameID>
12:   </saml:Subject>
13:   <saml:Conditions
14:     NotBefore="2005-01-31T12:00:00Z"
15:     NotOnOrAfter="2005-01-31T12:10:00Z">
16:   </saml:Conditions>
17:   <saml:AuthnStatement
18:     AuthnInstant="2005-01-31T12:00:00Z" SessionI       ="67775277772">
19:     <saml:AuthnContext>
20:       <saml:AuthnContextClassRef>
21:         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
22:       </saml:AuthnContextClassRef>
23:     </saml:AuthnContext>
24:   </saml:AuthnStatement>
25: </saml:Assertion>
```

SAML assertion validity period

AuthN element describing the act of authentication at the identity provider

SAML AUTHN ASSERTION: EXAMPLE

```
 1: <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
 2:   Ve
 3:   I
 4:                                        SAML:2.0:nameid-format:entity>
 5:
 6:
 7:
 8:
 9:                                        1.1:nameid-format:emailAddress">
10:
11:
12:   </
13:   <saml:Conditions
14:     NotBefore="2005-01-31                0Z"
15:     NotOnOrAfter="2005-01-3              0:00Z">
16:   </saml:Conditions>
17:   <saml:AuthnStatement
18:     AuthnInstant="2005-01-31T12:         :00Z" SessionIndex="67775277772">
19:     <saml:AuthnContext>
20:       <saml:AuthnContextClassRef>
21:         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
22:       </saml:AuthnContextClassRef>
23:     </saml:AuthnContext>
24:   </saml:AuthnStatement>
25: </saml:Assertion>
```

Notice that
- Authentication itself is *not* part of SAML
- Statement refers to an authentication act that took place at a prior time

SAML ATTRIBUTE ASSERTION: EXAMPLE

```
 1: <saml:AttributeStatement>
 2:   <saml:Attribute
 3:     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
 4:     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
 5:     Name="urn:oid:2.5.4.42"
 6:     FriendlyName="givenName">
 7:     <saml:AttributeValue xsi:type="xs:string"
 8:       x500:Encoding="LDAP">John</saml:AttributeValue>
 9:   </saml:Attribute>
10:   <saml:Attribute
11:     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
12:     Name="LastName">
13:     <saml:AttributeValue
14:       xsi:type="xs:string">Doe</saml:AttributeValue>
15:   </saml:Attribute>
16:   <saml:Attribute
17:     NameFormat="http://smithco.com/attr-formats"
18:     Name="CreditLimit">
19:     xmlns:smithco="http://www.smithco.com/smithco-schema.xsd"
20:     <saml:AttributeValue xsi:type="smithco:type">
21:       <smithco:amount currency="USD">500.00</smithco:amount>
22:     </saml:AttributeValue>
23:   </saml:Attribute>
24: </saml:AttributeStatement>
```

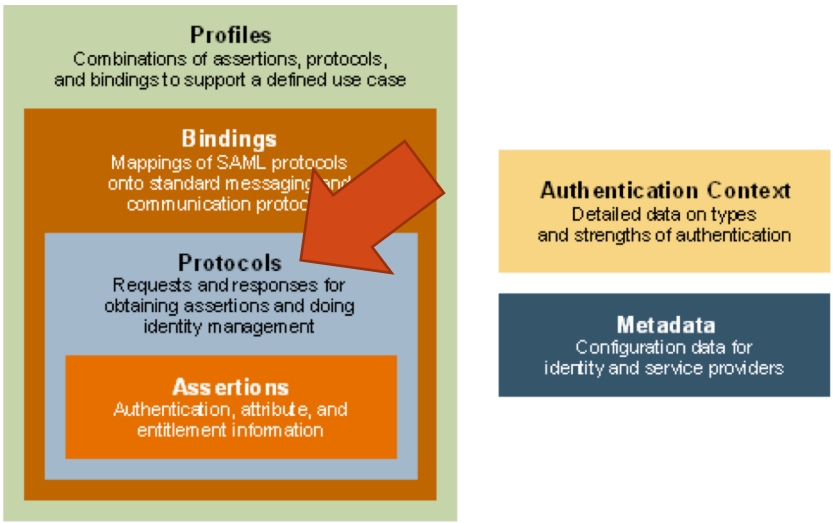
SAML ATTRIBUTE ASSERTION: EXAMPLE

```
1:  <saml:AttributeStatement>
2:    <saml:Attribute
3:       xmlns:x500="urn:oasis:names:tc:SAML:2
4:       NameFormat="urn:oasis:names:tc:SAML:
5:       Name="urn:oid:2.5.4.42"
6:       FriendlyName="givenName">
7:       <saml:AttributeValue xsi:
8:         x500:Encoding="LDAP"                xs:string
                                  ann</saml:AttributeValue>
9:    </saml:Attribute>
10:   <saml:Attribute
11:     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
12:     Name="LastName">
13:     <saml:AttributeValue
14:       xsi:type="xs:string">Doe</saml:AttributeValue>
15:   </saml:Attribute>
16:   <saml:Attribute
17:     NameFormat="http://smithco.com/attr                ts"
18:     Name="CreditLimit">
19:     xmlns:smithco="http://www.smithco.com/sm
20:     <saml:AttributeValue xsi:type="smithco:t
21:       <smithco:amount currency="USD">500.00<
22:     </saml:AttributeValue>
23:   </saml:Attribute>
24: </saml:AttributeStatement>
```

Asserts that the enclosing assertion's subject is associated with attribute *attrib* with value *val*

The value of the attribute "LastName" associated to the assertion's subject is "Doe"

SAML OVERVIEW



PROTOCOLS

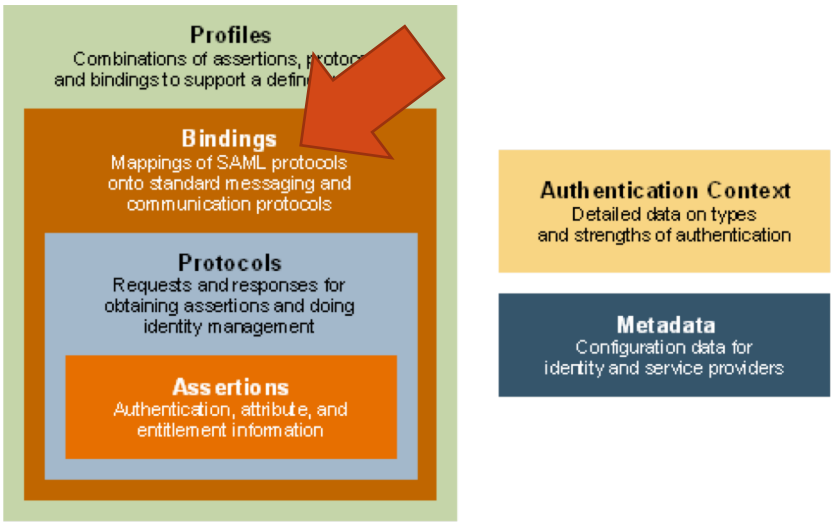
- Flow of assertion query and request
  - For obtaining SAML assertions
- Authentication request
- Artifact resolution
  - A mechanism by which protocol messages may be passed by references
- Single logout
- … and much more…

- **Communication protocol** = system that allows two or more entities to exchange information by defining the rules, syntax, semantics and synchronization of communication
- **Cryptographic protocol** = usually a small communication (i.e. its specification is typically quite compact, e.g., few exchange of messages in given formats) **designed to secure communication** (various security goals) by **using cryptographic primitives** (e.g., ciphers, hash functions, …)
- Typical security goal of SAML protocols is authentication of users, namely

The SP authenticates the user through an IdP assertion

Important remark
The behavior of a protocol is typically independent of how it is to be implemented

SAML OVERVIEW



BINDINGS

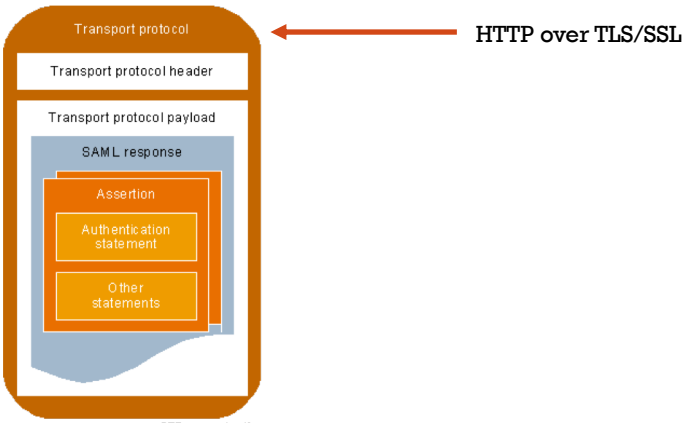
- The Hypertext Transfer Protocol (**HTTP**) is used to load web pages using links
- HTTP is an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack
- A typical flow over HTTP involves a client machine making a **request** to a server, which then sends a **response** message

- SAML requestors and responders communicate by exchanging messages
- The mechanism to transport these messages is called a *SAML binding*
- Types: SAML URI. SAML SOAP, …, **HTTP redirect**, HTTP POST, HTTP artifact
  - HTTP redirect **enables SAML protocol messages to be transmitted within URL parameters**
  - It enables SAML requestors and responders to communicate by using an HTTP user agent as an intermediary
  - The intermediary might be necessary if the communicating entities do not have a direct path of communication
  - The intermediary might also be necessary if the responder requires interaction with a user agent, such as an authentication agent.
  - HTTP redirect is sometimes called **browser redirect** in single sign-on operations
    - This profile is selected **by default**

Unique Resource Locator (URL): Protocol hostname File **https://www.fbk.eu/en/about-fbk/**

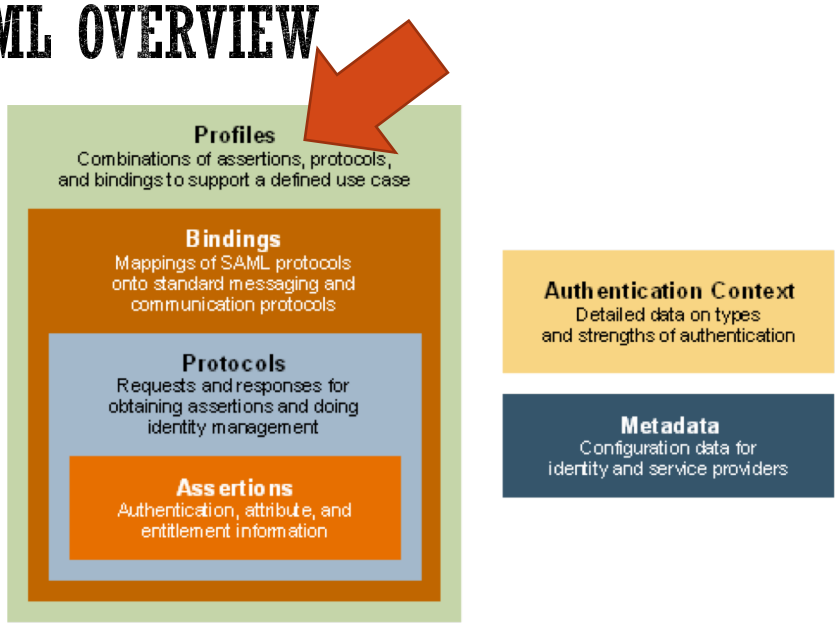
28

HOW ASSERTIONS ARE EXCHANGED



29

SAML OVERVIEW



S. Ranise - Security & Trust (FBK)

30

PROFILES

- SSO enables access to applications and resources within a single security domain
- Federated SSO enables SSO to applications across multiple security domains

- SAML 2.0 profiles combine protocols, assertions, and bindings **to create a federation and enable federated single sign-on**
- Types of profiles: **Web browser SSO**, **Single Logout**, Artifact resolution, …
  - **Web browser single sign-on** profile provides options regarding the initiation of the message flow and the transport of the messages:
    - Flow initiation: The message flow can be initiated from the identity provider or the service provider
    - Bindings: HTTP redirect, HTTP POST, HTTP artifact
  - **Single Logout** profile is used to terminate all the login sessions currently active for a specified user within the federation
    - A user who achieves single sign-on to a federation establishes sessions with more than one participant in the federation
    - The sessions are managed by a session authority, which in many cases is an identity provider
    - When the user wants to end sessions with all session participants, the session authority can use the single logout profile to globally terminate all active sessions

S. Ranise - Security & Trust (FBK)

Recall from slide 3:

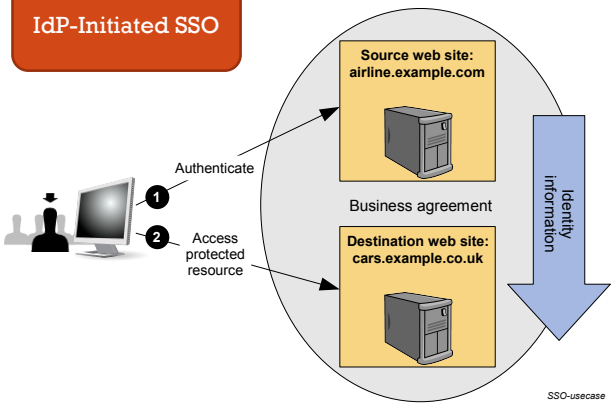
Security domain: an application or collection of applications trusting a common security token for authentication, authorization or session management

A **security token** is issued to users after they have actively authenticated with their identifiers and credentials (e.g., passwords or other authentication factors) to the security domain

WEB SSO PROFILE: 1ST SCENARIO

1. a user has a login session (that is, a *security context)* on a web site (*airline.example.com*) and is accessing resources on that site.
2. At some point, either explicitly or transparently, he is directed over to a partner's web site (*cars.example.co.uk*)
   - We assume that a federated identity for the user has been previously established between *airline.example.com* and *cars.example.co.uk* based on a business agreement between them.
3. The identity provider site (*airline.example.com*) asserts to the service provider site (*cars.example.co.uk*) that the user is known (by referring to the user by their federated identity), has authenticated to it, and has certain identity attributes (e.g. has a "Gold membership").
4. Since *cars.example.co.uk* trusts *airline.example.com*, it trusts that the user is valid and properly authenticated and thus creates a local session for the user

IdP-Initiated SSO



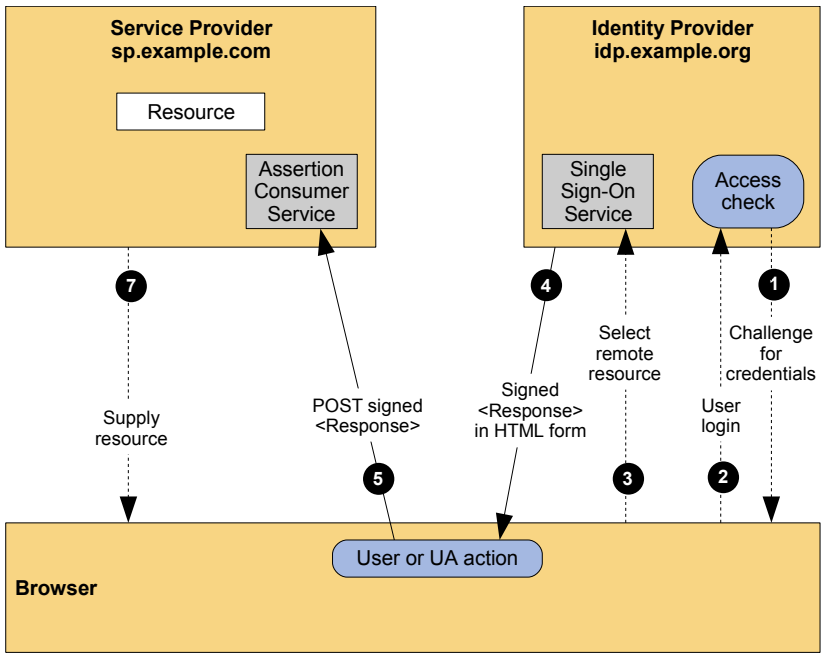
S. Ranise - Security & Trust (FBK)

32

WEB SSO 1

IdP-Initiated SSO

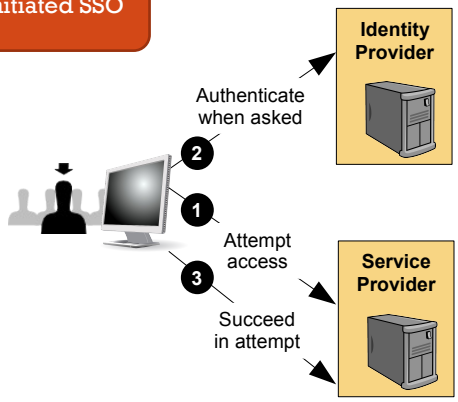
1. User is challenged to supply credentials to the IdP site
2. User provides valid credentials and a local logon security context is created
3. User selects a menu option or link on the IdP to request access to SP web site; this causes the IdP's SSO Service to be called
4. SSO Service builds a SAML assertion representing the user's logon security context
5. Browser issues an HTTP POST request to send a form to the SP's Assertion Consumer Service
6. Final access check to allow/deny user access to resource



WEB SSO PROFILE: 2ND SCENARIO

- **More common scenario** starts with a user visiting an SP site, possibly first accessing resources that require no special authentication or authorization
- When they subsequently attempt to access a protected resource at the SP, the SP will send the user to the IdP with an authentication request in order to have the user log in
- Once logged in, the IdP can produce an assertion that can be used by the SP to validate the user's access rights to the protected resource

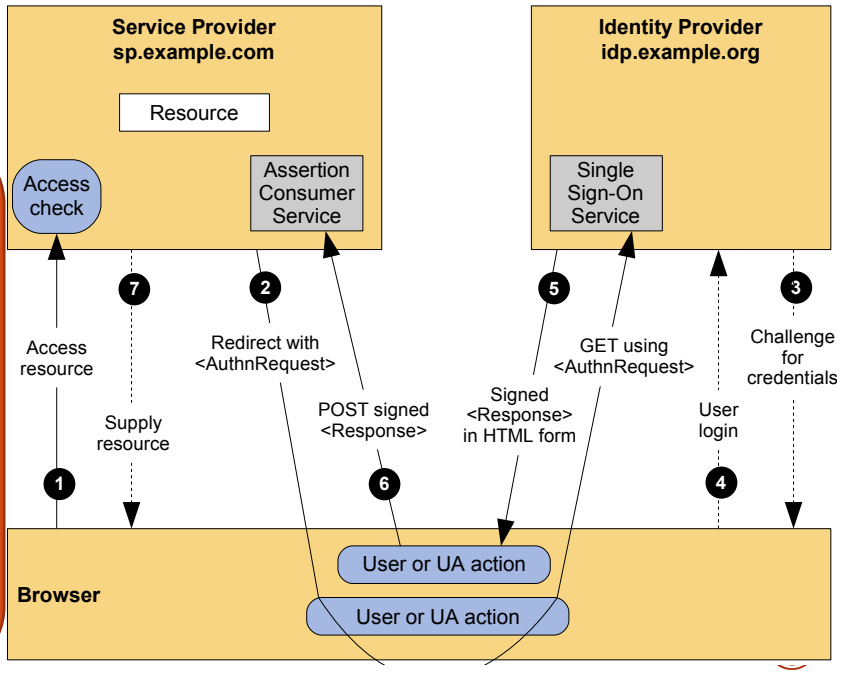
SP-Initiated SSO



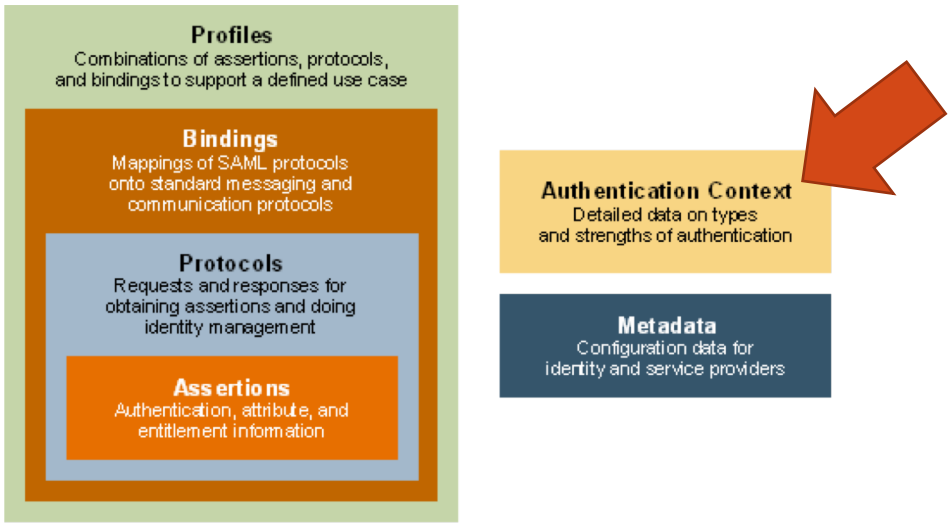
WEB SSO 2

SP-Initiated SSO

1. User attempts to access a resource on SP
2. SP sends a redirect response
3. SSO Service determines whether user has an existing logon security context at the identity provider; if not, IdP interacts with user to provide valid credentials
4. User provides valid credentials and a local logon security context is created
5. IdP Single Sign-On Service builds a SAML assertion representing the user's logon security context
6. The browser issues an HTTP POST request to send the form to the SP's Assertion Consumer Service
7. Final access check to allow/deny user access to resource



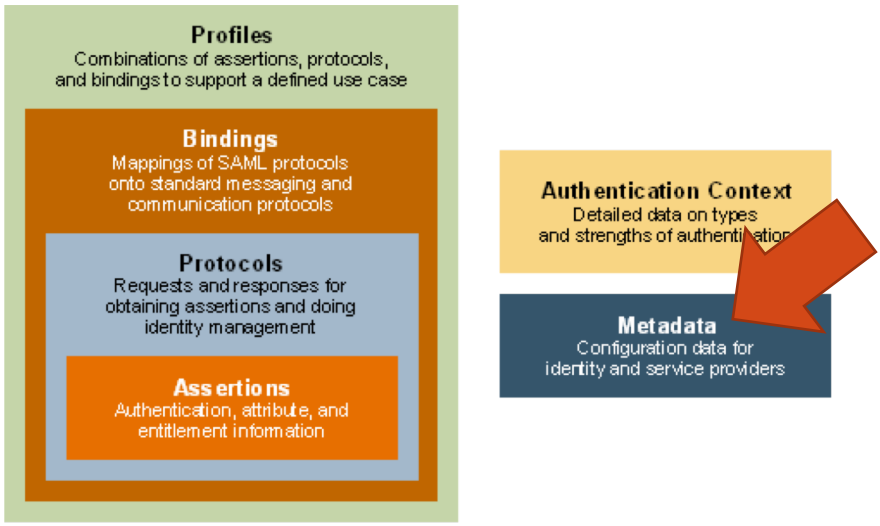
SAML OVERVIEW



AUTHENTICATION CONTEXT

- It indicates **how a user authenticated at an Identity Provider**
- The Identity Provider includes the authentication context in an assertion at the request of a Service Provider or based on configuration at the Identity Provider
- A Service Provider can require information about the authentication process to establish a level of confidence in the assertion before granting access to resources
- Motivation
  - Existing SAML federation deployments have adopted a "**Levels Of Assurance**" (or **LOA**) model for categorizing the wide variety of authentication methods into a small number of levels, typically **based on some notion of the strength of the authentication**
  - Service Providers then decide which level of assurance is required to access specific protected resources, based on some assessment of "value" or "risk"

SAML OVERVIEW



METADATA (1)

- A SAML metadata document describes a SAML deployment such as a SAML identity provider or a SAML service provider
- Deployments share metadata to establish a baseline of **trust and interoperability**
- Minimum set of metadata to be shared:
  - **Entity ID**
  - **Cryptographic Keys**
  - **Protocol Endpoints** (bindings and URLs)
- Every SAML system entity has an entity ID, a globally unique identifier used in software configurations, relying-party databases, and client-side cookies
  - On the wire, every SAML protocol message contains the entity ID of the issuer.
- For authentication purposes, a SAML message may be digitally signed by the issuer
- To verify the signature on the message, the message receiver uses a public key known to belong to the issuer
- Similarly, to encrypt a message, a public encryption key belonging to the ultimate receiver must be known to the issuer
- In both situations—signing and encryption—trusted public keys must be shared in advance

METADATA (2)

- Once the message is signed and encrypted, the issuer sends the message to a trusted protocol endpoint, the location of which must be known in advance
- Upon receipt, the message receiver decrypts the message (using its own private decryption key) and verifies the signature (using a trusted public key in metadata) before mapping the entity ID in the message to a trusted partner
- This scenario requires each party to know the other in advance
- To establish a baseline of trust, parties share metadata with each other
- Initially, this may be as simple as sharing information via email
- Over time, as the number of SAML partners grows, the natural tendency is to automate the metadata sharing process
- An implementation that supports SAML Web Browser SSO requires a schema-valid SAML metadata file for each SAML partner

41

SAML: SOME SECURITY CONSIDERATIONS

SAML SECURITY (1)

- Just providing assertions from an asserting party to a relying party may not be adequate to ensure a secure system
  - How does the relying party trust what is being asserted to it?
  - What prevents a "**man-in-the-middle**" attack that might grab assertions to be illicitly "**replayed**" at a later date?
- SAML defines a number of security mechanisms to detect and protect against such attacks
  - Primary mechanism is for the relying party and asserting party to have a pre-existing trust relationship which typically relies on a **Public Key Infrastructure** (PKI)
  - While use of a PKI is not mandated by SAML, it is recommended

SAML SECURITY (2)

- Use of particular security mechanisms are described for each SAML binding in the standard
- General recommendations are the following
  - Where **message integrity & confidentiality** are required, then SSL/TLS is recommended
  - When a relying party requests an assertion from an asserting party, **bi-lateral authentication** is required and the use of SSL/TLS using mutual authentication is recommended
  - When a response message containing an assertion is delivered to a relying party **via a user's web browser** (for example using the HTTP POST binding), then to ensure **message integrity**, it is mandated that the response **message be digitally signed** using XML Signature

SAML SECURITY (3)

Some more attacks…

- Message expiration
  - SAML messages should contain a timestamp of when the request was issued, when it expires or both
  - If the SAML message never expires or if the expiration is not verified, there is a greater risk of a message falling into the hands of an attacker
  - Check the message for timestamps
- Message replay
  - Assertions should contain a unique ID that is only accepted once by the application
- SAML from Different Recipient
  - An application should only accept a SAML message intended for the SP application
  - If the application does not perform this check, it may accept a SAML message generated from authenticating to another application and allow an attacker into the application as the user from the other application
- XML External Entity (XXE)
  - A SAML message is just a user-provided XML message that is processed by the Service Provider
  - Check all standard XML attack vectors such as XXE that forces to parse malicious data

SAML PRIVACY

- Privacy refers to both
  - a **user's ability to control how their identity data is shared & used** and
  - to mechanisms that **inhibit** their **actions** at multiple service providers from **being inappropriately correlated**
- SAML has a number of mechanisms that support deployment in privacy
  - **Persistent pseudonyms** established between an identity and a service provider
    - pseudonyms do not themselves enable inappropriate correlation between service providers (as would be possible if the identity provider asserted the same identifier for a user to every service provider, a so-called *global* identifier)
  - ***one-time*/transient identifiers** ensure that every time a certain user accesses a given service provider through a SSO operation from an identity provider, that service provider will be unable to recognize them as the same individual as might have previously visited
  - **Authentication Context** allows a user to be authenticated at a sufficient (but not more than necessary) assurance level, appropriate to the resource they may be attempting to access at some service provider
  - SAML allows the claimed fact of a **user consenting to certain operations** to be expressed between providers. How, when or where such consent is obtained is out of scope for SAML

46

NATIONAL IDENTITY INFRASTRUCTURES

Examples: Italian digital identity solutions

- SPID = Sistema Pubblico Identità Digitale (Public System for Digital Identity)
- CIE 3.0 = Carta d'Identità Elettronica 3.0 (Electronic Identity Card)

47

SPID (SISTEMA PUBBLICO IDENTITÀ DIGITALE)

Public System for Digital Identity

SPID: OVERVIEW

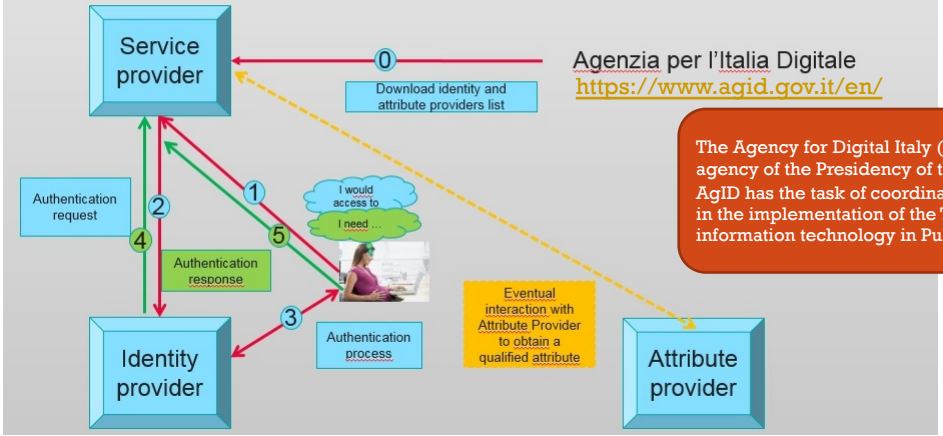
Based on the SAML Web Browser SSO Profile

Based on SAML 2.0

SPID – Identification and security levels

https://www.agid.gov.it/en/platforms/spid

Public System for Digital Identity



Agenzia per l'Italia Digitale
https://www.agid.gov.it/en/

The Agency for Digital Italy (AgID) is the technical agency of the Presidency of the Council of Ministers. AgID has the task of coordinating public administrations in the implementation of the Three-Year Plan for information technology in Public Administration.

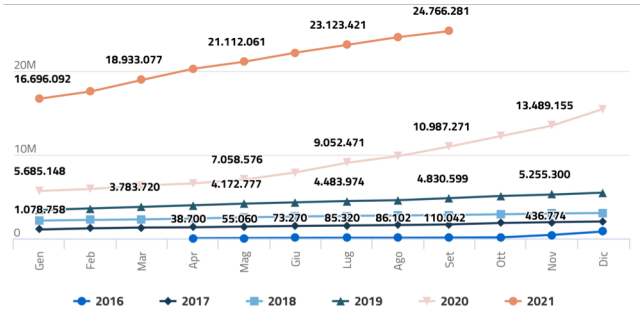
ABOUT SPID

- Identity providers are private
  - Do you trust them?
  - Phylosophical/polytical question
- Legal provisions sometime get in the way
  - Ex: Assurance levels
  - **SSO seems impossible when considering higher assurance levels**
- User identification for enrollment is delicate
  - Sometime ago, an italian journalist "hacked" the procedure and successfully obtained the digital identity of a colleague…
  - http://www.ilfattoquotidiano.it/2016/11/04/identita-digitale-ce-un-buco-nella-sicurezza-cosi-ti-divento-matteo-renzi/3093093/
- Adoption seems to lag behind although with the pandemics increased substantially
  - Situation: see graph on the right

Assurance Levels

- ~~Level 0: little or no confidence in asserted identity's validity~~
- Level 1: some confidence in asserted identity's validity
- Level 2: high confidence in asserted identity's validity
- Level 3: very high confidence in asserted identity's validity

https://avanzamentodigitale.italia.it/it/progetto/spid



SPID REGISTER (1)

Metadata management in SPID

- Repository of all the information related to the entities adhering to the SPID and represents the evidence of the so-called **circle of trust** established therein
- The relationship of trust on which the federation established in SPID is based is achieved through **the intermediation of the Agency, third party guarantor, through the process of accreditation of digital identity providers, the attribute authorities and service providers**
- Adhesion to SPID constitutes the establishment of a relationship of trust with all existing members accredited by the Agency, based on the sharing of the standard security levels of assurance declared and guaranteed by SPID
- Adhesion to the trust agreement among member entities is demonstrated by the presence of such entities in the SPID Register managed by the Agency

SPID REGISTER (2)

Metadata management in SPID

- The *federation registry* contains the list of entities that have passed the accreditation process and are therefore part of the SPID federation
- For each entity the registry contains an entry called **AuthorityInfo** consisting of:
  - SAML identifier of the entity
  - name of the subject to which the federation entity refers
  - type of entity (Identity Provider, Attribute Authority, Service Provider)
  - URL of the metadata provider service
  - List of qualified attributes which can be certified by an Attribute Authority
- The federation registry is populated by AgID following stipulation of the agreements and updated by said Agency during the activities related to management of the agreements and the supervision of the parties of the SPID circuit

52

CIE 3.0 (CARTA D'IDENTITÀ ELETTRONICA)

Electronic identity card

STORED PERSONAL INFORMATION

- Name
- Surname
- Place and date of birth
- Residency
- Holder's picture
- Two fingerprints (one of each hand), only if the applicant is aged 12 or over
  - Can be accessed only by law enforcement agencies
- Different validity periods
  - 10 years for adults aged 18 and above
  - 5 years for minors aged 3–18
  - 3 years for children aged up to 3

CAPABILITIES

- NFC = Near Field Communication
  - set of communication protocols for communication between two electronic devices over a distance of at most 4 cm
  - low-speed connection with simple setup
  - NFC devices can act as electronic **identity documents** and **keycards**
  - Used in contactless payment systems and allow mobile payment replacing or supplementing systems such as credit cards
- Cryptography
  - AES (192, 256 bits) and Triple DES (112 bits)
  - SHA-2 or SHA-1
  - Diffie-Hellman 2048 bits and Elliptic Curve Diffie-Hellman 192 bits
  - RSA 2048 bits (v1.5)
  - X.509 certificate with personal data signed by official CA
  - …

Only some combinations are supported

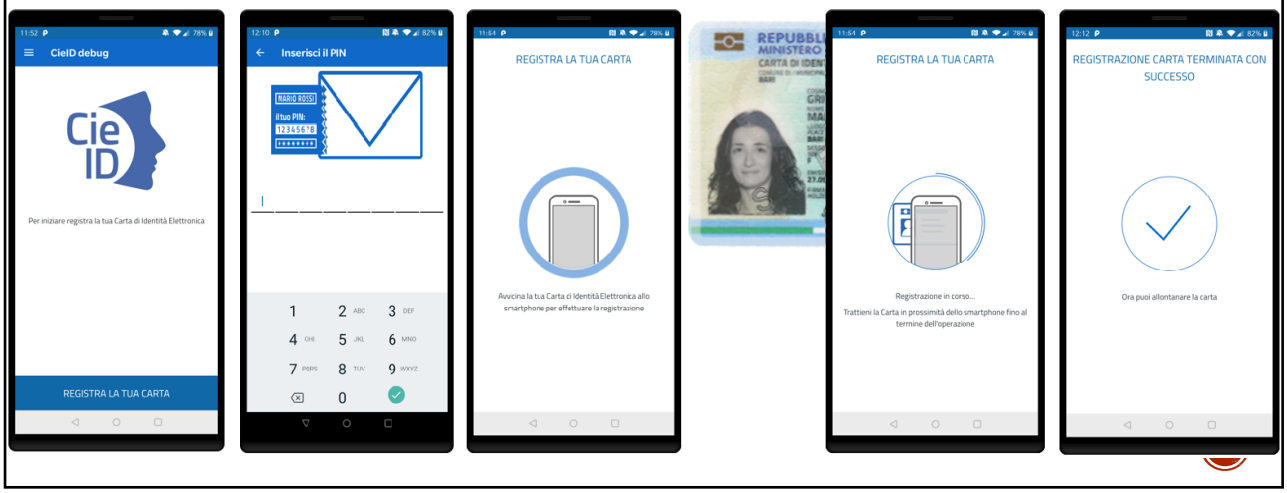
55

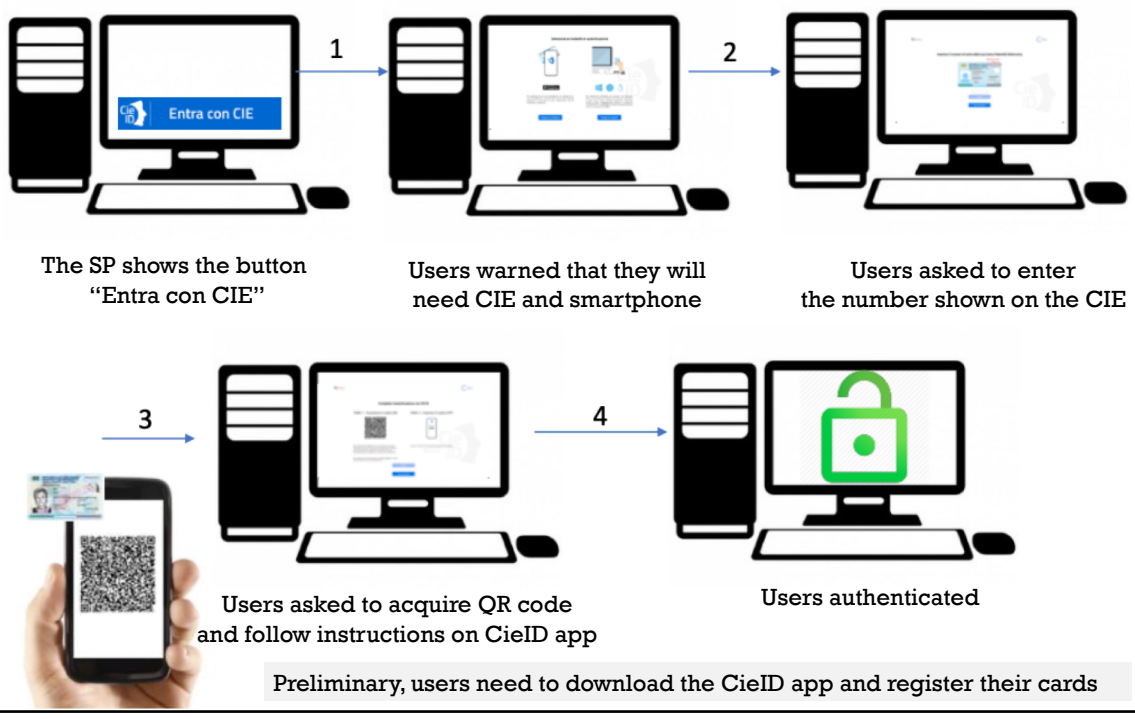
A POSSIBLE USE OF CIE 3.0

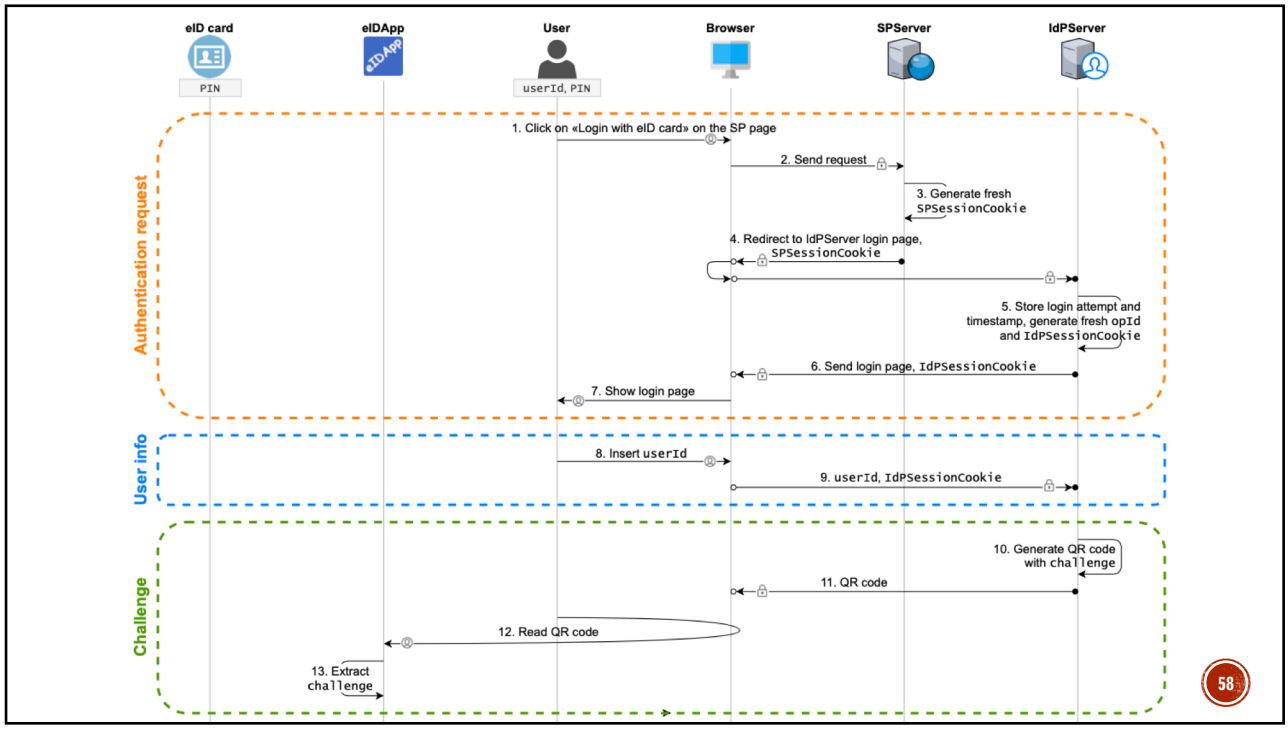
Compliant with the SAML Web Browser SSO Profile

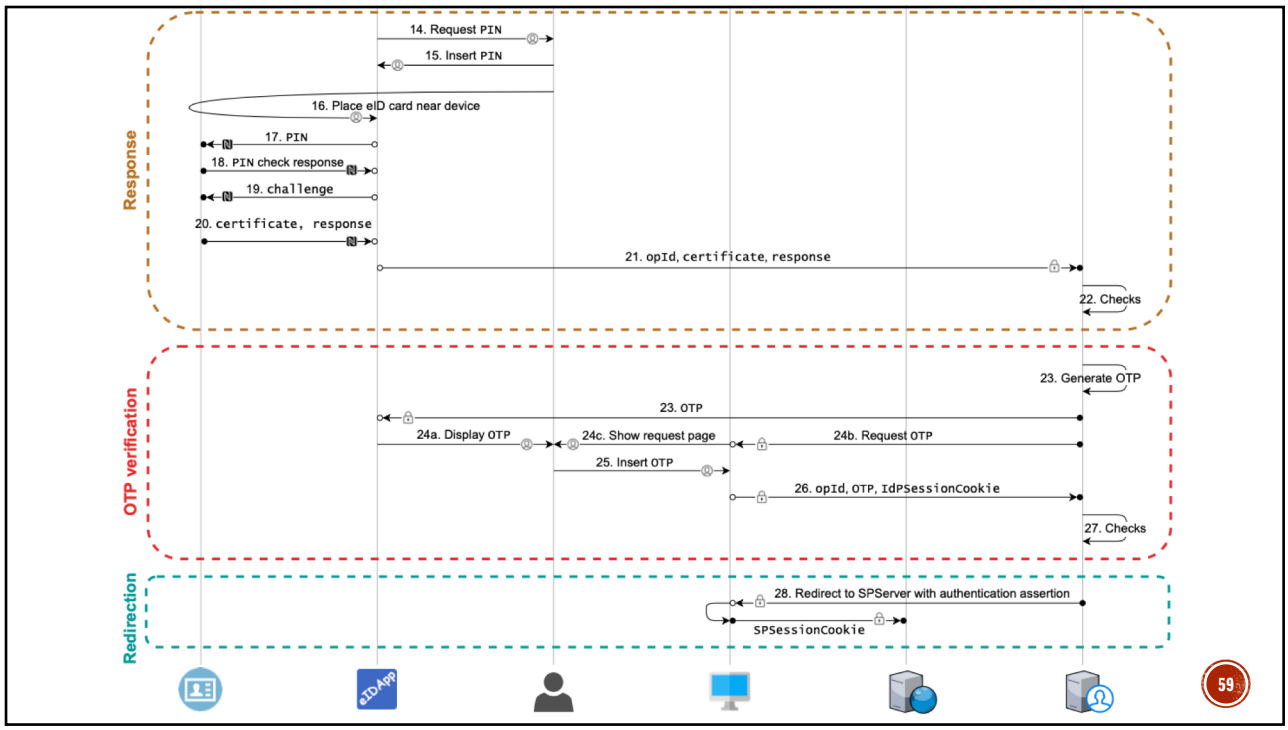
Behind the scenes of the button 

CIE REGISTRATION ON CIEID APP






eID card
eIDApp
User
Browser
SPServer
IdPServer
PIN
userId, PIN
Authentication request
1. Click on «Login with eID card» on the SP page
2. Send request
3. Generate fresh SPSessionCookie
4. Redirect to IdPServer login page, SPSessionCookie
5. Store login attempt and timestamp, generate fresh opId and IdPSessionCookie
6. Send login page, IdPSessionCookie
7. Show login page
User info
8. Insert userId
9. userId, IdPSessionCookie
Challenge
10. Generate QR code with challenge
11. QR code
12. Read QR code
13. Extract challenge
58

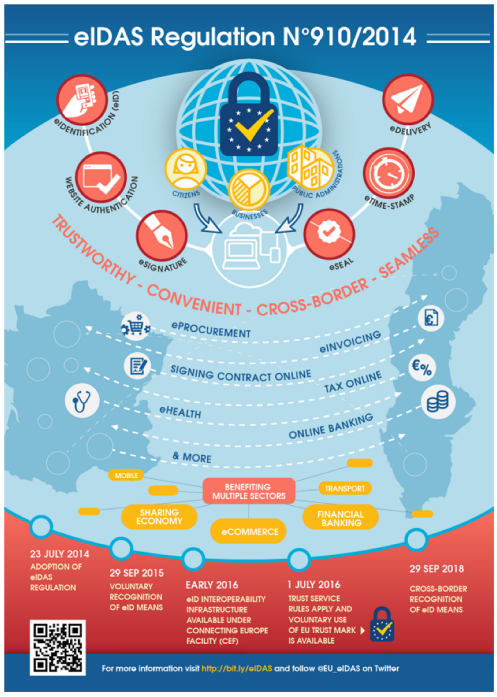

Response
14. Request PIN
15. Insert PIN
16. Place eID card near device
17. PIN
18. PIN check response
19. challenge
20. certificate, response
21. opId, certificate, response
22. Checks
OTP verification
23. Generate OTP
23. OTP
24a. Display OTP
24c. Show request page
24b. Request OTP
25. Insert OTP
26. opId, OTP, IdPSessionCookie
27. Checks
Redirection
28. Redirect to SPServer with authentication assertion
SPSessionCookie
59

60

EUROPEAN IDENTITY INFRASTRUCTURE

Aka the portability of national digital identities of Member States and more…

eIDAS

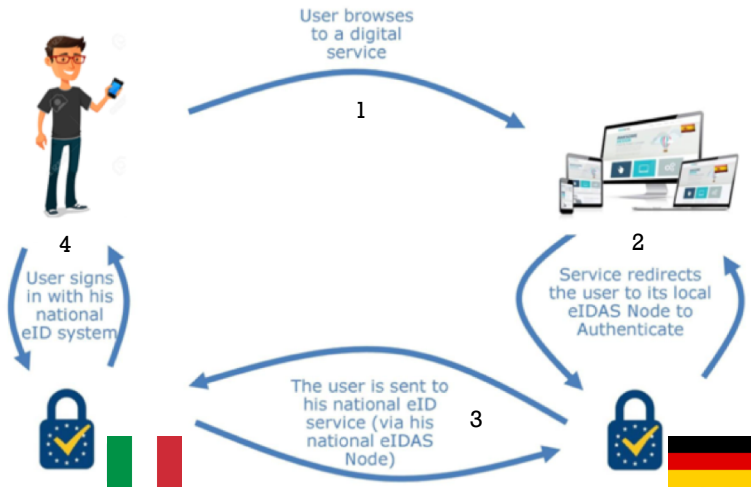


SPID is interoperable with eIDAS!

CIE is also interoperable with eIDAS!

61

eIDAS EXAMPLE: OPENING BANK ACCOUNT

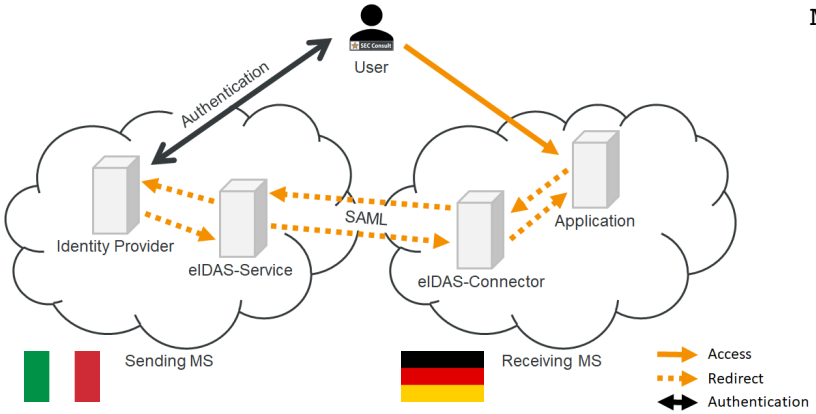


EIDAS OVERVIEW

- If an Italian citizen wants to authenticate against a German online service, first the German eIDAS-Node (eIDAS-Connector) is directed by the web application to initiate the authentication process
- It sends a request to the Italian eIDAS-Node (eIDAS-Service)
- The Italian eIDAS-Node forwards the user to a system that is equipped to authenticate the Italian citizen using the national eID scheme
- After authentication, the German eIDAS-Connector receives the citizen's information which it forwards to the web application
- **eIDAS relies on SAML for communication between the eIDAS-Connector and eIDAS-Service** (called eIDAS-Nodes)

HOW EIDAS WORKS

MS=Member State



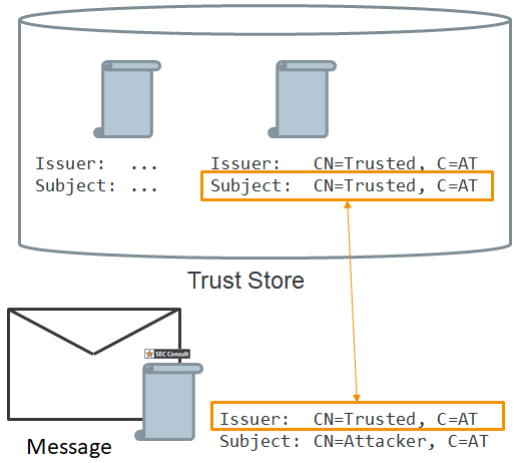
EIDAS VULNERABILITY (2019)

- As each Member State must provide its own eIDAS-Node, the European Commission has provided the **eIDAS-Node Integration Package**, which can be used as a basis for implementing such a service
- Focus on the SAML parsing code
- SEC Consult found a vulnerability that basically allowed attackers to bypass the signature verification, allowing athem to send any SAML message to an affected eIDAS-Node
- The attacker could therefore, e.g., send a **manipulated SAML response to an eIDAS-Connector to authenticate as anybody**
- The vulnerable code was used to verify the trust of the certificate the SAML response was signed with:
  1. The certificate is accepted if it is in the local trust store
  2. Otherwise, **the issuer certificate of the entity certificate is retrieved from either the local trust store or from the supplemental certificates in the SAML message**
  3. If a trust path can be established between the issuer certificate and a certificate in the trust store, the entity certificate is accepted

https://sec-consult.com/en/blog/2019/10/vulnerability-in-eu-cross-border-authentication-software-eidas-node/

EIDAS VULNERABILITY (CONT'D)

- It was found that, in step 2, the application searches for the the issuer certificate by comparing the Issuer DN of the entity certificate to the Subject DN of the potential issuer certificates
- **The application does not verify whether the entity certificate has been correctly signed by the issuer certificate**
  - Other checks, such as whether the basic constraints of the issuer certificate allow it to act as a certificate issuer are not verified
- An attacker can therefore sign a manipulated SAML response with a **forged certificate**
- The certificate must contain an Issuer DN that matches the subject of a certificate in the trust store
  - The subject must contain the country of the citizen (e.g. CN=FAKE, C=AT).



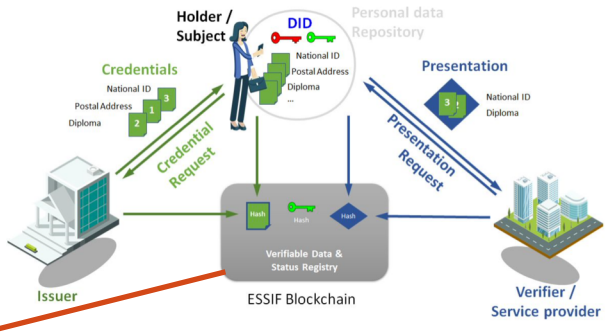
EIDAS CURRENTLY UNDER DISCUSSION (1)

- Focus on digital wallets
- A personal digital identity wallet is a **user-controlled app** empowering the citizens to provide proofs of their identity or attributes so that the citizen is able to selectively share personal information with services by fishing (verifiable) credentials from the wallets
- The Wallet could integrate various credentials asserting attributes obtained from private or public providers, in addition to the national eID
  - Examples of credentials: driver's licenses, health certificates, insurance, education, profession, etc
- More details available at https://ec.europa.eu/commission/presscorner/detail/en/ip_21_2663

EIDAS CURRENTLY UNDER DISCUSSION (2)

- **Self-sovereign identity** = users control the verifiable credentials that they hold and their **consent** is required to use those credentials
- **Decentralized Identifiers** = identify subjects that the controller decides that they identify
- **Verifiable credentials** = digital equivalent of physical credentials such as credit cards, passports, driving licenses, qualifications and awards
  - EU is creating an eIDAS compatible **European Self-Sovereign Identity Framework (ESSIF)** making use of decentralized identifiers and the European Blockchain Services Infrastructure (EBSI)

Challenge: data interoperability



https://www.eesc.europa.eu/sites/default/files/files/1._panel_-_daniel_du_seuil.pdf

TAKEAWAYS

- SAML allows service providers to outsource identity management and focus on their core business
  - Reduced administration burden
  - Improved interoperability, usability, security and privacy
- SAML profiles are useful use case scenarios
  - Web SSO most widely adopted
  - SSO has increasing importance and is gaining wider and wider adoption
- SAML is ideal starting point to build infrastructures for digital identity management
  - Key enablers in an increasing digital world (SPID, eIDAS)
  - First line of defense against attackers

RECAP QUESTIONS

- What are the goals of SAML?
- What is the structure of a SAML assertion?
- What is a SAML profile? Give an example of SAML profile.
- What is the difference between IdP-Initiated and SP-Initiated Web SSO?
- What is the flow of an IdP-Initiated Web SSO?
- What is the flow of an SP-Initiated Web SSO?
- What are the main security concerns underlying the deployment of SAML? What are the main mitigations measures?
- What is SPID? What is eIDAS? Is there a relationship between the two?
- Give an example of scenario in which eIDAS is useful.