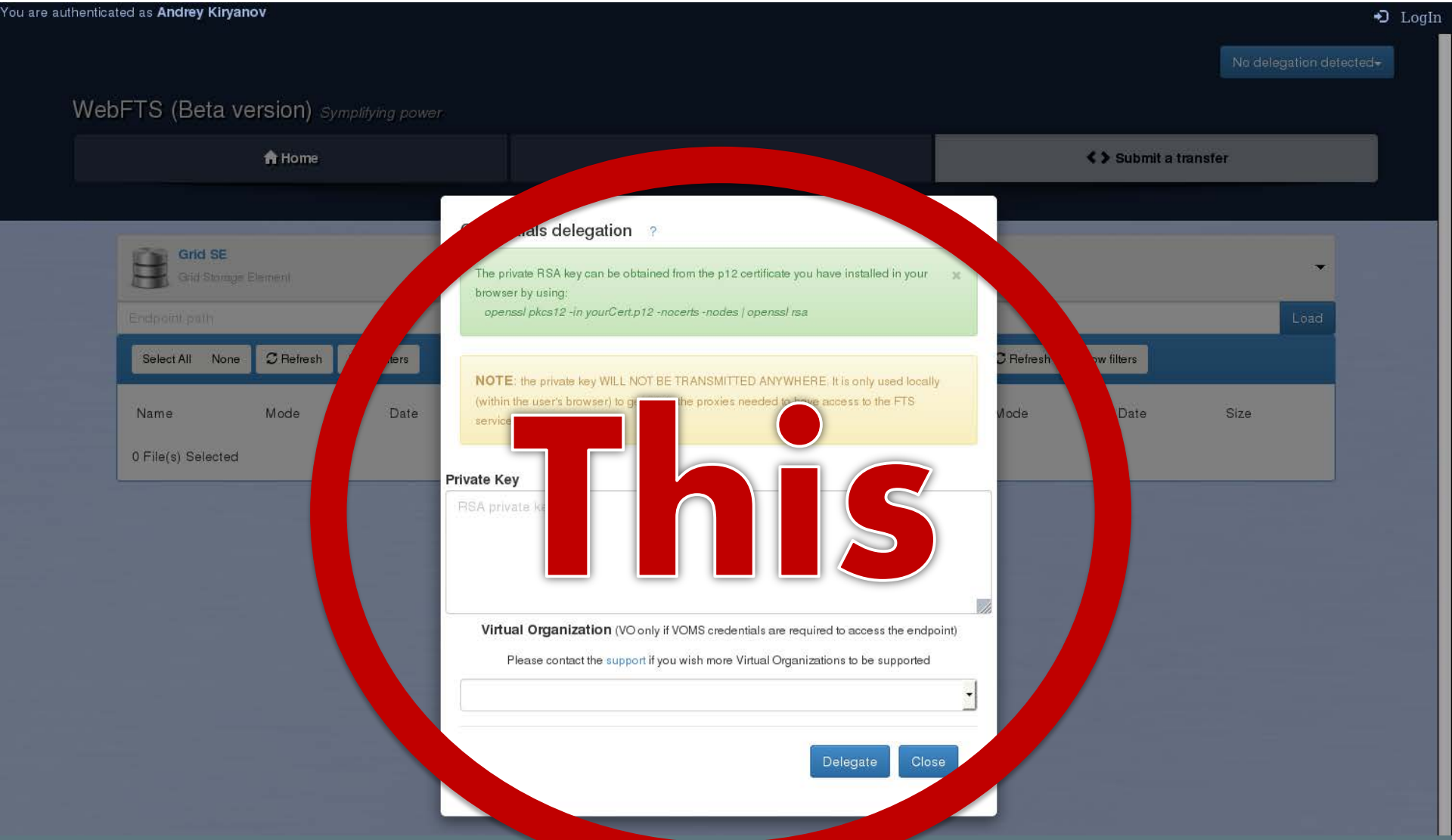
The saga of WebFTS and Federated Identity

Andrey Kiryanov
IT/SDC

15/12/2014

The Reason:


You are authenticated as Andrey Kiryanov
LogIn
No delegation detected
WebFTS (Beta version) Symplifying power
Home
Submit a transfer
Grid SE
Grid Storage Element
Endpoint path
Load
Select All
None
Refresh
Name
Mode
Date
Size
0 File(s) Selected
ials delegation
The private RSA key can be obtained from the p12 certificate you have installed in your browser by using:
openssl pkcs12 -in yourCert.p12 -nocerts -nodes | openssl rsa
NOTE: the private key WILL NOT BE TRANSMITTED ANYWHERE. It is only used locally (within the user's browser) to g the proxies needed to have access to the FTS service
Private Key
RSA private ke
Virtual Organization (VO only if VOMS credentials are required to access the endpoint)
Please contact the support if you wish more Virtual Organizations to be supported
Delegate
Close
This

What is a Federated Identity?

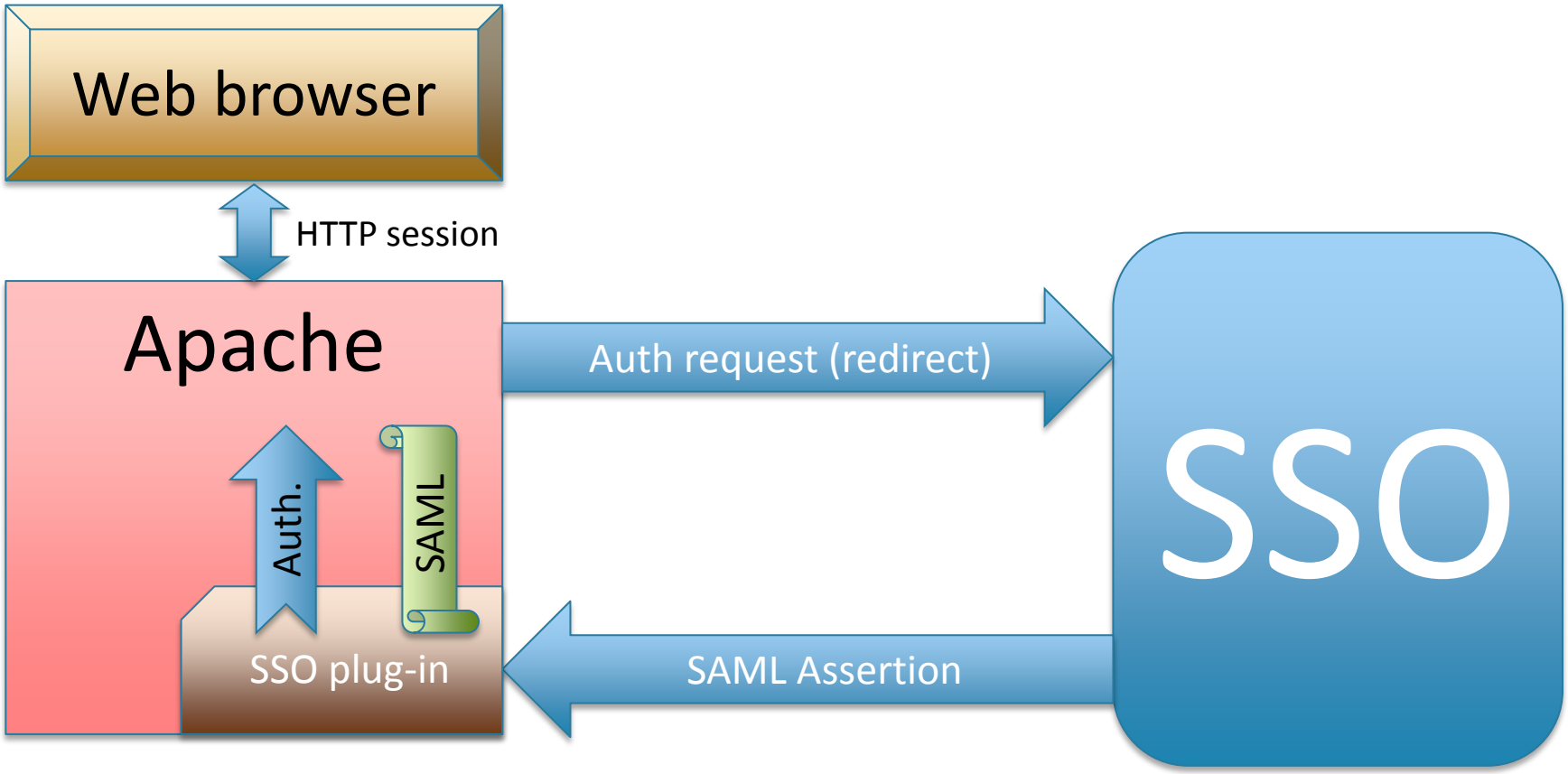
- It is the means of linking a person's electronic identity and attributes, stored across multiple distinct identity management systems (from Wikipedia).

- In our case it works like “extended SSO” where you can log-in on other sites (like INFN) with credentials from your home site (like CERN). You only have to register once in your home organization (Identity Provider – IdF IP) and all other sites (Service Providers – IdF SPs) will recognize you.

How is it implemented?

- Magic + tons of XML
- CERN SSO service is based on Microsoft's ADFS (Active Directory Federation Services)
- In order to benefit from SSO your web server (Apache) needs a special plug-in:
  - Shibboleth – supported by CERN, widespread solution, supports all possible standards, a real pain in the ~~a~~head to configure.
  - Mellon – pure SAML2 SP. Distribution from EPEL does not work with ADFS, but the one from Git does. Easy to configure, will be supported (may be even recommended) by CERN in the near future.

What happens when you log-in to SSO?



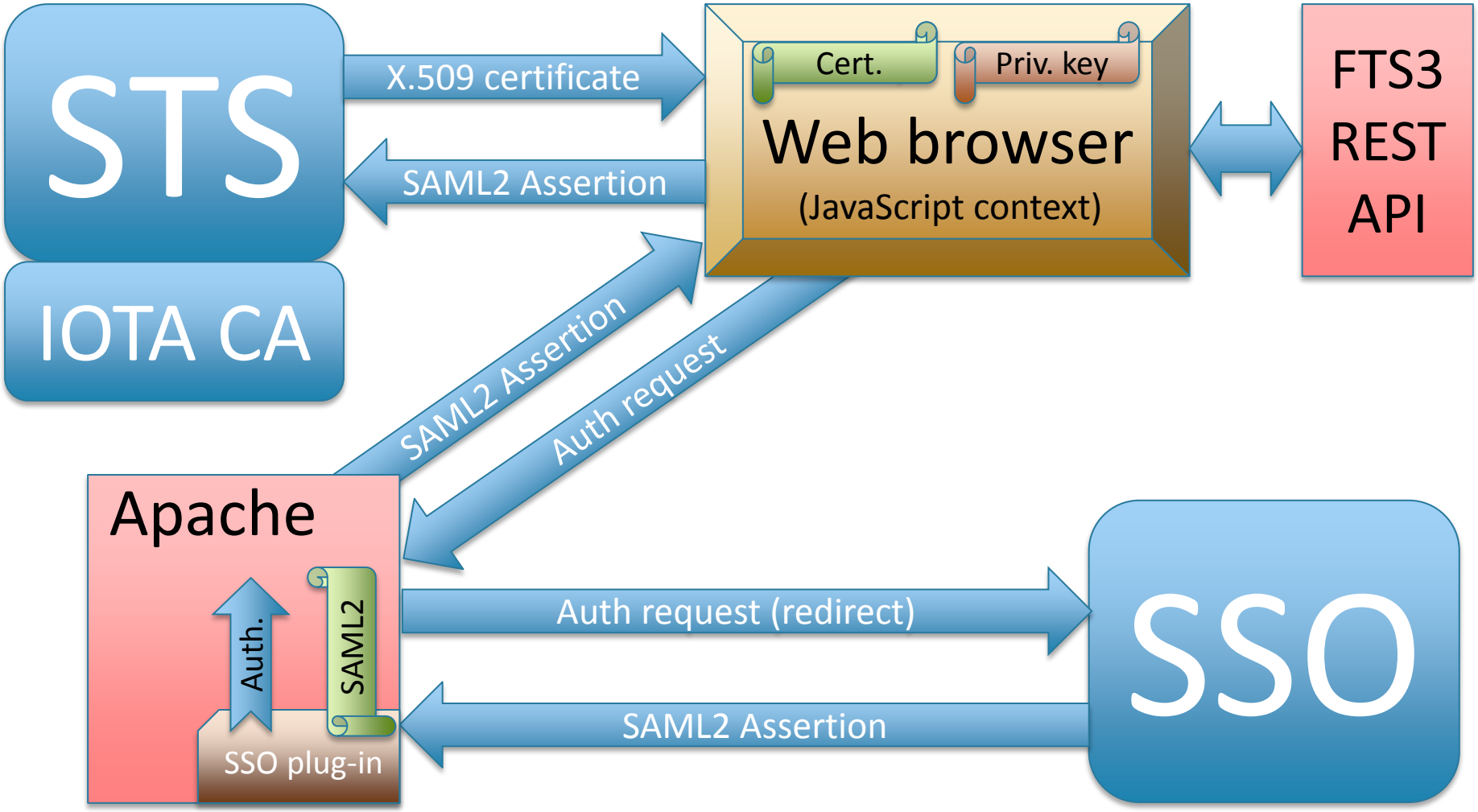
SAML = Security Assertion Markup Language

SAML Assertion is essentially a signed list of attributes (name, email, etc.)

STS – the new guy

- Security Token Service (STS) consumes SAML2 assertions and produces X.509 credentials in return.
  - STS is an implementation of WS-Trust OASIS standard and it speaks SOAP.
- This functionality is based on so-called IOTA CA (Identifier-Only Trust Assurance Certification Authority) that issues short-living (days) X.509 certificates.
  - At CERN we can get such certificates from “CERN CA” (which is NOT “CERN Grid CA”) – the same that signs EduRoam certificates.

What's in all this for WebFTS?



Changes made

- On FTS3 REST API side we had to implement a new authorization scheme.
  - It uses standard HTTP Authorization header with user certificate and timestamp signed by a private key.
  - Otherwise there's no way to use public and private keys from JavaScript context with AJAX.
- On WebFTS side we provided JavaScript functions that cover interaction with STS and handling of SAML2 assertions.
  - STS is not CORS-aware (so far). In order to access it with AJAX we had to use Apache reverse-proxy functionality to map STS into WebFTS namespace. In fact, we can do that for REST API as well ☺

Next steps

- Above all we have to convince sites to trust IOTA-profile CAs.
- The way STS issues certificates has to change. Basically STS has more than one mode of operation:
  - It can generate a key pair, sign it with a CA, and send both certificate and private key back to us. This is what is used right now, but this is wrong because private key is transmitted over the network.
  - It can generate a proxy certificate (with or without VOMS extensions) based on a public key provided from our side. This is more secure because we generate a key pair by ourselves and private key stays on our side, but this will require changes in the delegation code on WebFTS side.
- VOMS integration is not yet finished (not even started AFAIK) on STS side. We have to wait until it's available for testing.
- We have to think of the way we associate different identities of the same user (e.g. normal X.509 certificate and IdF).

What we have achieved so far?

- IdF-enabled WebFTS is a working prototype available at https://webfts-dev.cern.ch/
- This is an important step towards “X.509-free” access to Grid resources.
- The same technology may be used for other types of services, e.g. job submission.
- We have a deep understanding of the fact that IdF is still a work in progress, so do not expect stable and polished solutions in the nearest future.

Thank you!

(demo time)