

# Enhancing Cyber Security in Tango Controls: A Special Interest Group Workshop

# MAXIV

## Introduction

Anton Joubert

# Acknowledgements

8th Control System Cyber-Security Workshop (CS)2/HEP @ ICALEPCS 2023

<https://indico.cern.ch/event/1270052/>

EPICS Collaboration Meeting in April 2023

<https://indico.fnal.gov/event/58280/>

Authors: George McIntyre, Gregory White, Michael Davidsaver, Leonce Mekinda

# Agenda

Background

Karabo – European XFEL

EPICS

What about Tango?

Common issues

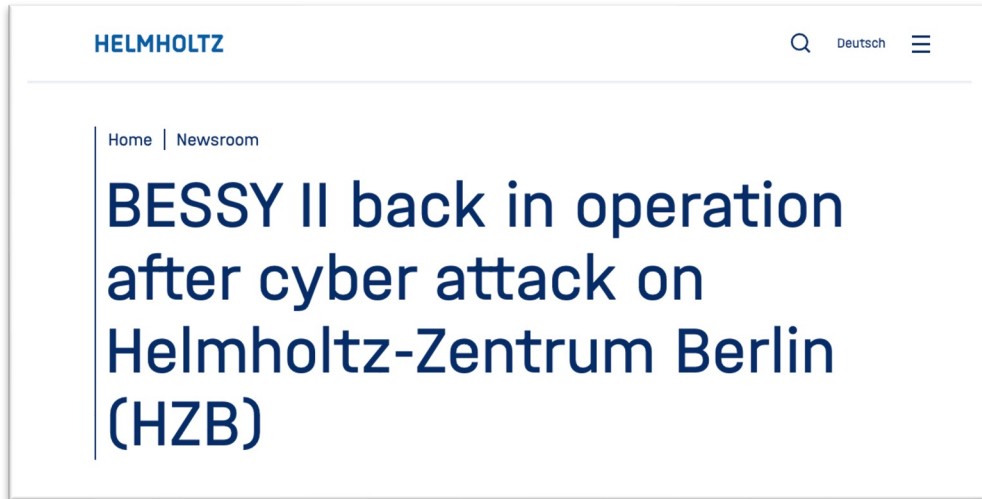
Use cases

Conclusion

# Background

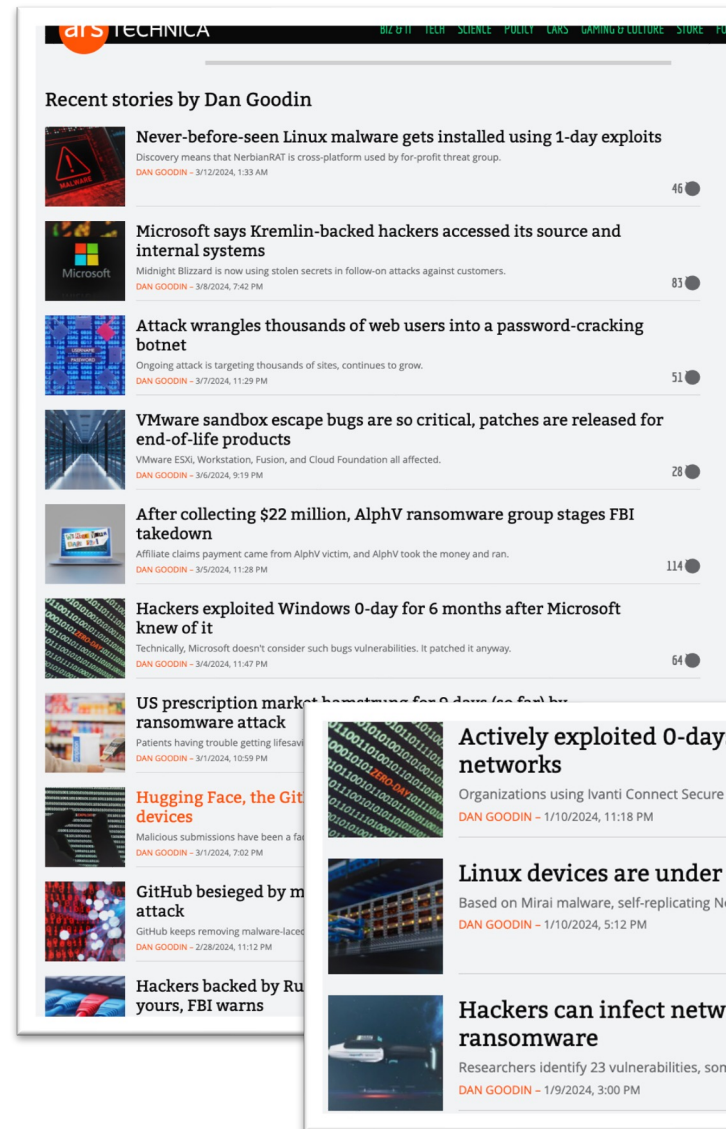


# Security news



July 2023

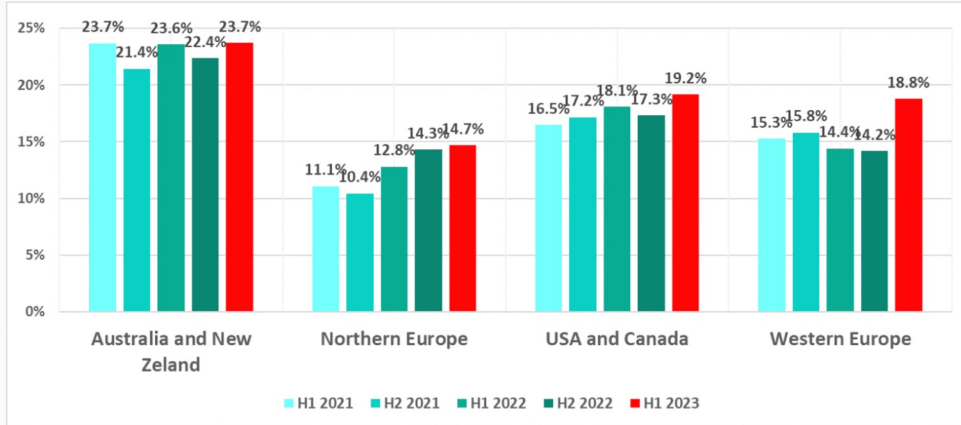
Source: <https://www.helmholtz.de/en/newsroom/bessy-ii-back-in-operation-after-cyber-attack-on-helmholtz-zentrum-berlin-hzb/>



36 stories in 2024

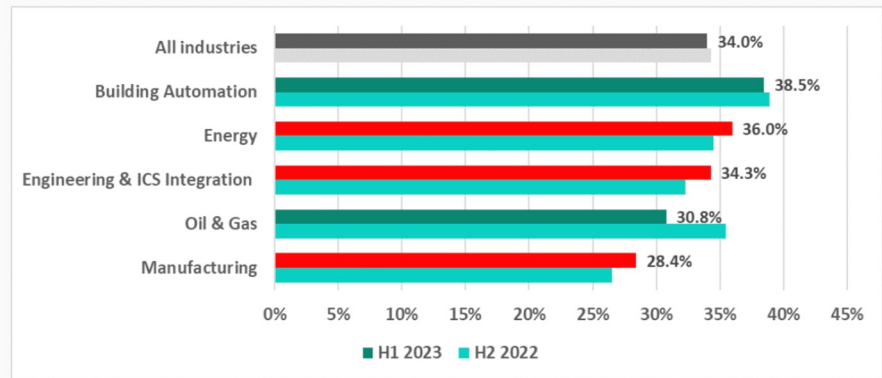
Source: <https://arstechnica.com/author/dan-goodin/>

# Kaspersky study



Percentage of ICS computers on which malicious objects were blocked in selected regions

ICS: Industrial Control System



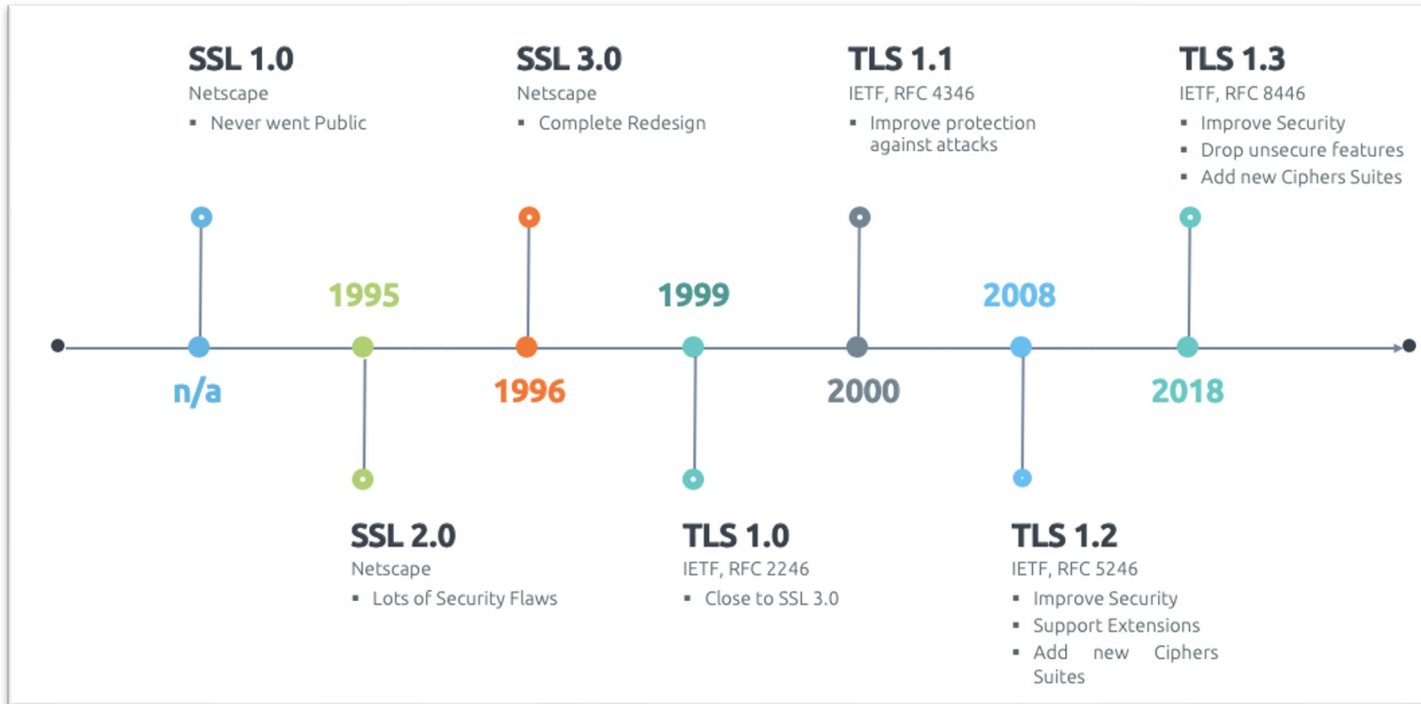
Percentage of ICS computers on which malicious objects were blocked in selected industries

Source: [https://www.kaspersky.com/about/press-releases/2023\\_attacks-on-industrial-sector-hit-record-in-second-quarter-of-2023](https://www.kaspersky.com/about/press-releases/2023_attacks-on-industrial-sector-hit-record-in-second-quarter-of-2023)

# Glossary

<b>AAA</b>	Authentication Authorisation Accounting
<b>Authentication</b>	Are you really you? I.e., identity
<b>Authorisation</b>	What are you allowed to do? I.e., privileges
<b>Accounting</b>	Who did what when? aka Auditing
<b>PKI</b>	Public Key Infrastructure. The way certificates and public-private key pairs are managed, implemented and operated.
<b>CA</b>	Certificate Authority. Trusted third party. The entity in a PKI that is responsible for issuing public-key certificates and exacting compliance.
<b>Certificate</b>	A set of data that uniquely identifies a public key (which has a private key) and an owner that is authorized to use the key pair. Digitally signed by a CA.
<b>SSL</b>	Secure Sockets Layer. Encryption-based Internet security protocol. Predecessor to TLS.
<b>TLS</b>	Transport Layer Security . Authentication and encryption protocol. HTTPS = HTTP over TLS

# Web security timeline



## Recommended configurations

Mozilla maintains three recommended configurations for servers using TLS. Pick the correct configuration depending on your audience:

- **Modern:** Modern clients that support TLS 1.3, with no need for backwards compatibility
- **Intermediate:** Recommended configuration for a general-purpose server
- **Old:** Services accessed by very old clients or libraries, such as Internet Explorer 8 (Windows XP), Java 6, or OpenSSL 0.9.8

Recommendation source: [https://wiki.mozilla.org/Security/Server\\_Side\\_TLS](https://wiki.mozilla.org/Security/Server_Side_TLS)

Image source: Boris Rogier <https://www.networkdatapedia.com/post/3-things-you-should-know-about-https-ssl-tls-traffic-with-wireshark>

# Web certificates

Digital certificates aka  
X.509 certificates aka  
PKI certificates.

Provided by global *Certificate Authorities*.  
E.g., Let's Encrypt, Verisign, Thawte, Entrust, etc.

Or can "self-sign" with limitations.

Read more:

<https://www.keyfactor.com/education-center/what-is-pki/>  
<https://www.feistyduck.com/library/bulletproof-tls-guide/online/>  
<https://letsencrypt.org/>

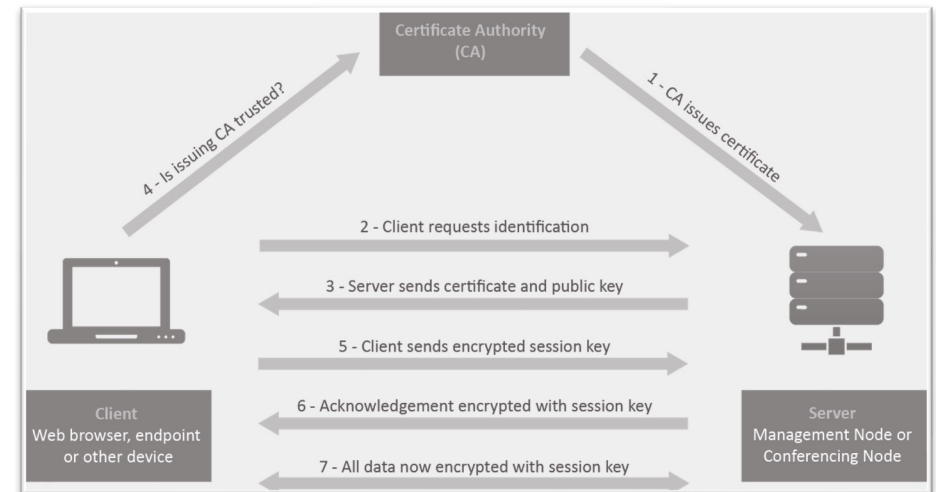
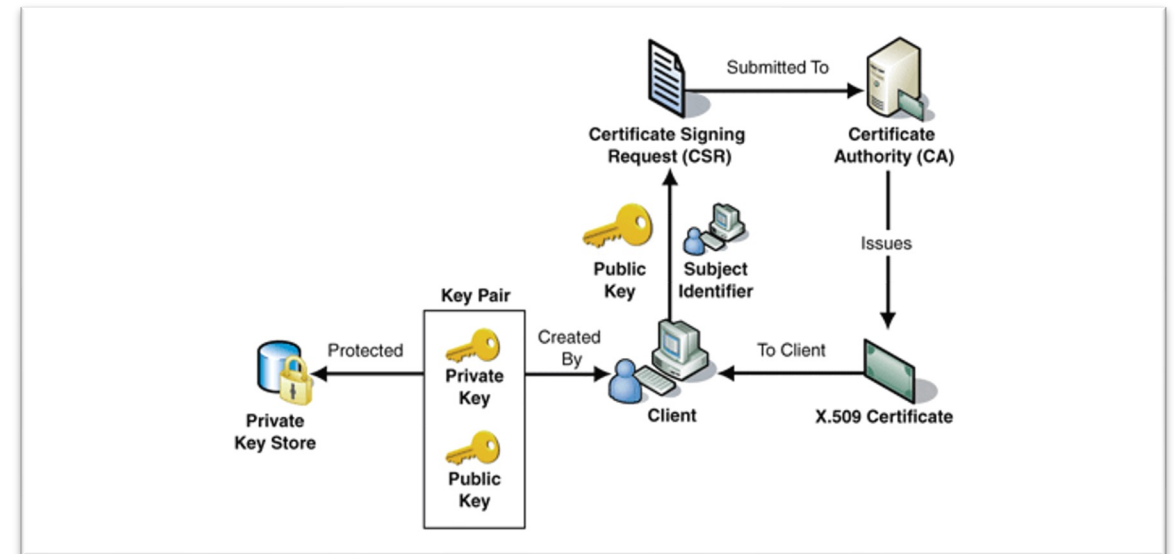


Image sources: The SSL Store: <https://www.thesslstore.com/blog/ssltls-certificate-its-architecture-process-interactions/>

Pexip AS: [https://docs.pexip.com/admin/certificate\\_management.htm](https://docs.pexip.com/admin/certificate_management.htm)



# Web authentication: SAML and SSO

SAML: Security Assertion Markup Language

SSO: Single Sign-On

IdP: Identity Provider

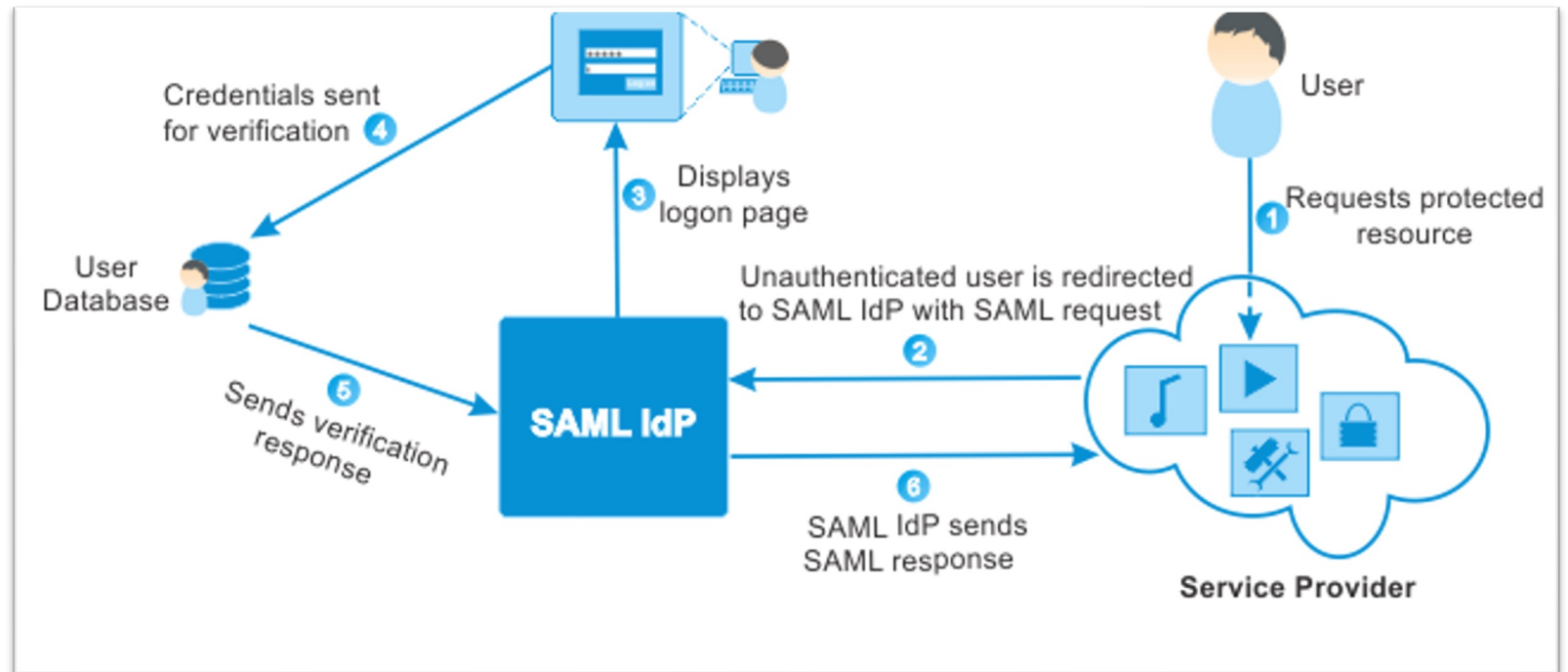
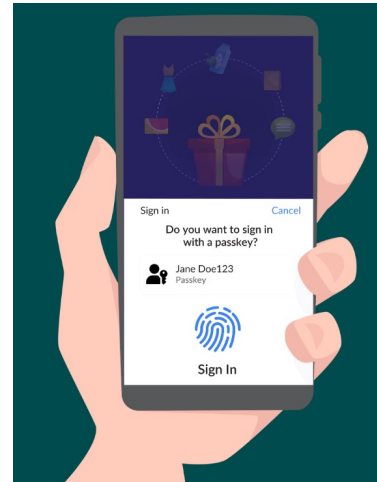
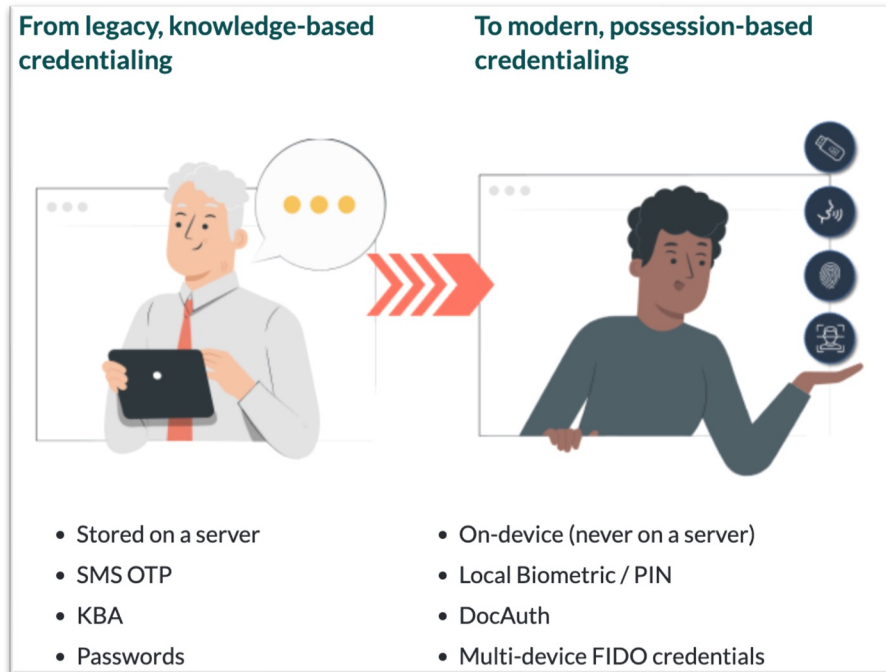


Image source: <https://bigdataanalyticsnews.com/how-does-saml-work/>

# Web authentication: WebAuthn and FIDO2



WebAuthn credentials also referred to as passkeys

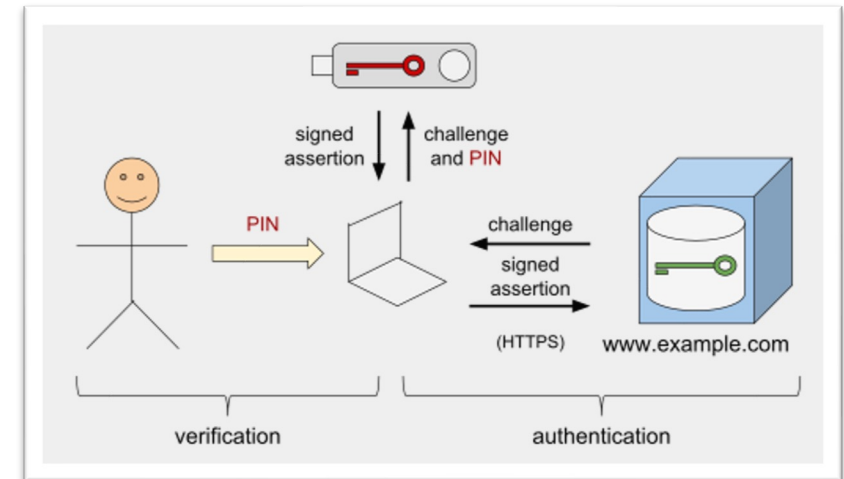


Image source: <https://fidoalliance.org/fido2/>

Image source: Tom Scavo [https://en.wikipedia.org/wiki/File:Passwordless\\_Web\\_Authentication.svg](https://en.wikipedia.org/wiki/File:Passwordless_Web_Authentication.svg)

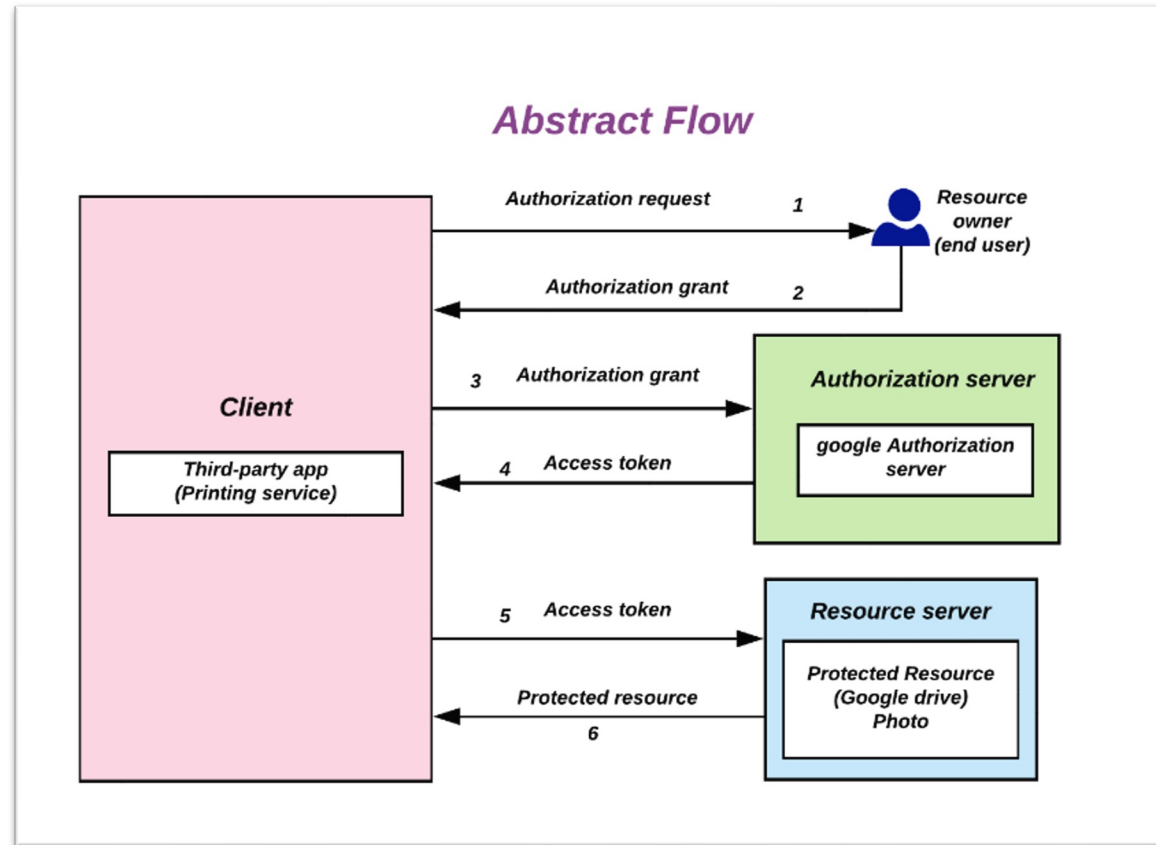
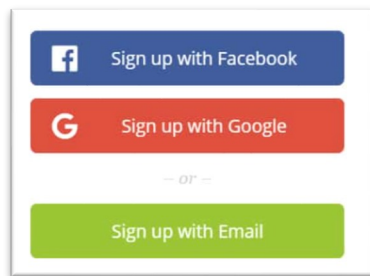
# Web authorisation: OAuth 2.0

OAuth: Open Authorisation

Specifies a *delegation* protocol

Can be misused as pseudo-authentication

*OpenID Connect*: combines authentication and authorisation on top of OAuth



E.g.,  
Single Sign-On (SSO)  
provider

Image source: Devansvd <https://en.wikipedia.org/wiki/File:Abstract-flow.png>

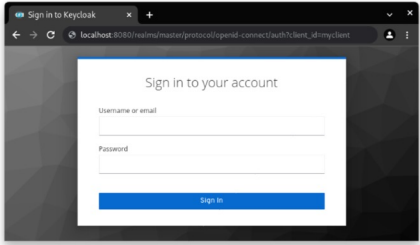
# Web tools: identity and access management

One example of an open-source implementation is Keycloak

**Single-Sign On**

Users authenticate with Keycloak rather than individual applications. This means that your applications don't have to deal with login forms, authenticating users, and storing users. Once logged-in to Keycloak, users don't have to login again to access a different application.


This also applies to logout. Keycloak provides single-sign out, which means users only have to logout once to be logged-out of all applications that use Keycloak.



**Identity Brokering and Social Login**

Enabling login with social networks is easy to add through the admin console. It's just a matter of selecting the social network you want to add. No code or changes to your application is required.

Keycloak can also authenticate users with existing OpenID Connect or SAML 2.0 Identity Providers. Again, this is just a matter of configuring the Identity Provider through the admin console.



**User Federation**

Keycloak has built-in support to connect to existing LDAP or Active Directory servers. You can also implement your own provider if you have users in other stores, such as a relational database.




Image source: <https://www.keycloak.org>

# Research facility control systems



# Karabo – European XFEL

# Securing Light Source SCADA Systems

ICALEPCS2017  
European XFEL

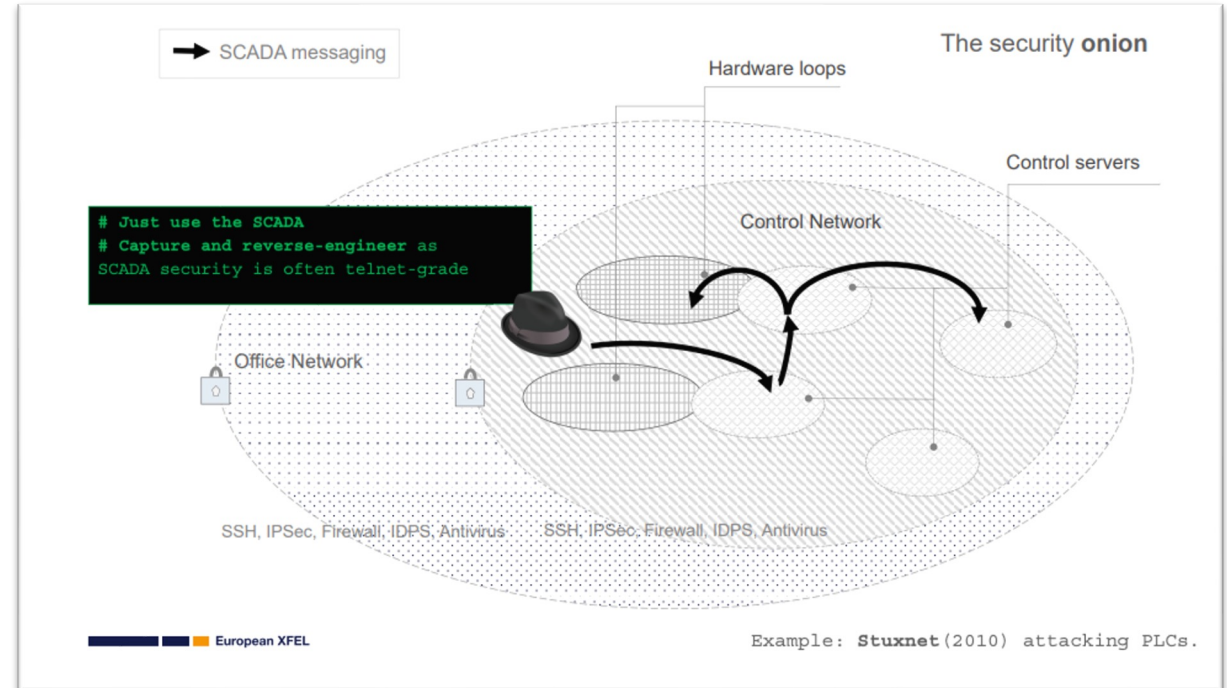
## Securing Light Source SCADA Systems

Leonce Mekinda, Valerii Bondar, Sandor Brockhauser,  
Cyril Danilevski, Wajid Ehsan, Sergey Esenov, Hans Fangohr, Gero Flucke, Gabriele Giovanetti, Steffen  
Hauf, David Gareth Hickin, Anna Klimovskaia, Luis Maia, Thomas Michelat, Astrid Muennich, Andrea  
Parenti, Hugo Santos, Kerstin Weger, Chen Xu.  
European XFEL GmbH

Barcelona, 12/10/2017

### Overview

- The security of SCADA systems is an increasing concern as they interconnect a significant number of COTS computers via IP networks; support *de facto* standards like USB.
- What happens once attackers have been granted access to / broke into the Control Network?
  - Can they do everything?
  - Can they easily escalate their privileges?
- "We trust whoever has access to the Control Network"
  - Would you let your personal laptop unlocked 24/7 in a control room? If no, why should the control system be less protected than your laptop?
- We suggest to secure the SCADA system beyond the general IT infrastructure security
  - Device servers would authenticate and authorize users for every issued message.

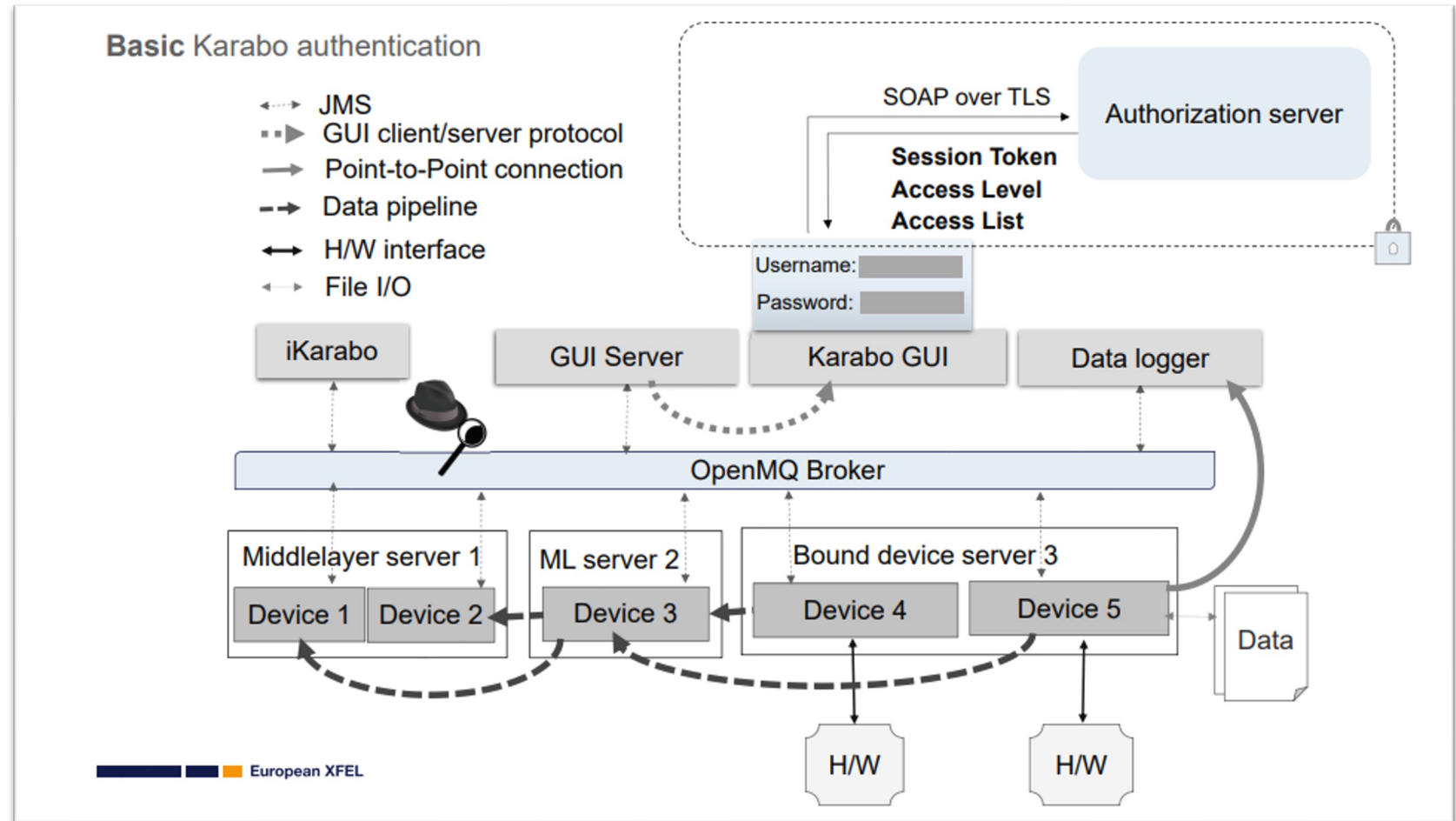


See details in the [paper](#)

Slide credit: Leonce Mekinda [https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02\\_talk.pdf](https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02_talk.pdf)

# Securing Light Source SCADA Systems

- Five Global access levels in Karabo:
  - Observer
  - User
  - Operator
  - Expert
  - Admin (required for example for interlock deactivation)
- Access exception list per device.



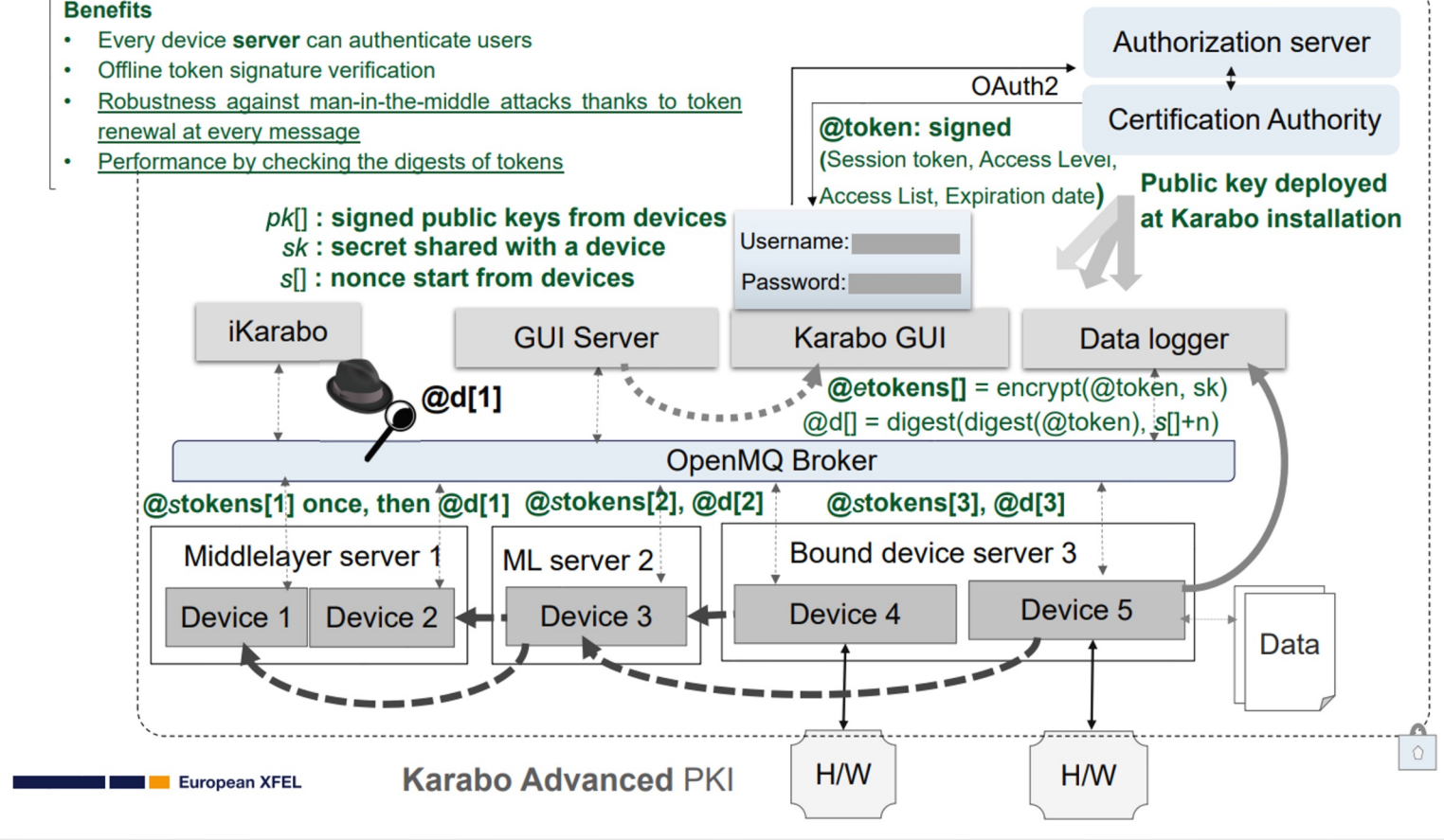
Slide credit: Leonce Mekinda [https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02\\_talk.pdf](https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02_talk.pdf)

# Securing Light Source SCADA Systems

- Five Global access levels in Karabo:
  - Observer
  - User
  - Operator
  - Expert
  - Admin (required for example for interlock deactivation)
- Access exception list per device.

## Benefits

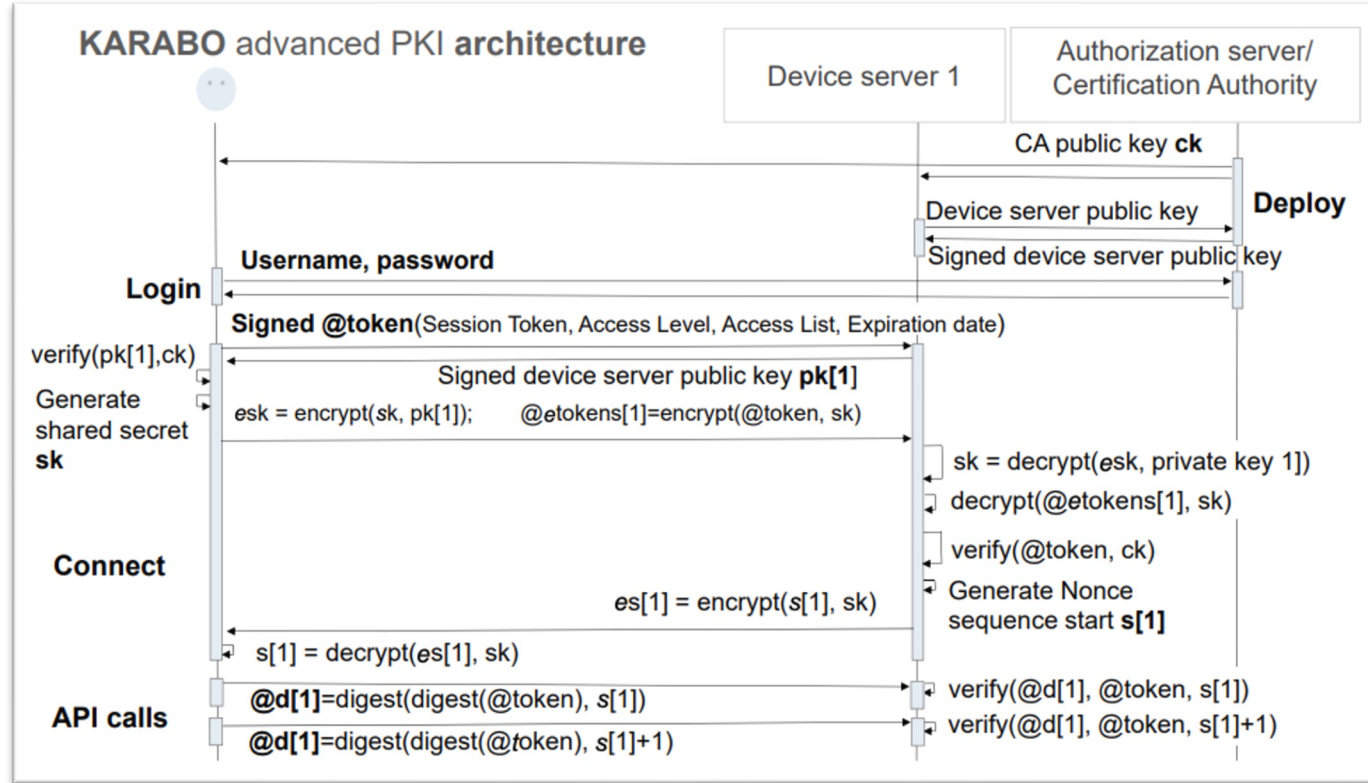
- Every device **server** can authenticate users
- Offline token signature verification
- Robustness against man-in-the-middle attacks thanks to token renewal at every message
- Performance by checking the digests of tokens



Slide credit: Leonce Mekinda [https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02\\_talk.pdf](https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02_talk.pdf)

# Securing Light Source SCADA Systems

- Five Global access levels in Karabo:
  - Observer
  - User
  - Operator
  - Expert
  - Admin (required for example for interlock deactivation)
- Access exception list per device.



<https://github.com/European-XFEL/Karabo> (note that it wasn't implemented this way)

Slide credit: Leonce Mekinda [https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02\\_talk.pdf](https://accelconf.web.cern.ch/icalepcs2017/talks/thbpa02_talk.pdf)



EPICS


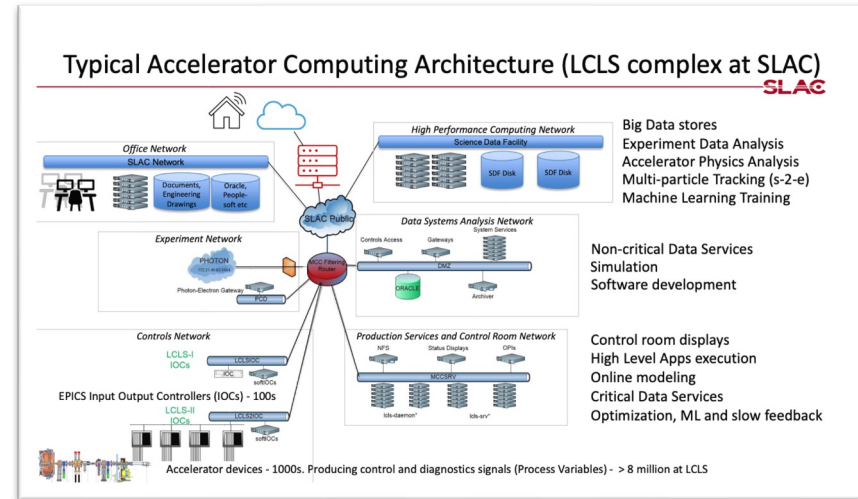
# Review from SLAC

**SLAC Initiatives on Accelerator Cyber Security**

Greg White,  
Prepared for EPICS Collaboration Meeting Spring 2023  
April 26, 2023

Greg White, for Erwin Lopez, Mark McCullough, Amedeo Perazzo, Ken Brobeck, Mark Foster, Daron Chabot, Mike Zelazny, Lance Nakata, Matt Gibbs, Andrea Chan, Arash Alavi, Poonam Pandi, Lisa Christiansen, Uy Chu, Syed Hasan

**Many Thanks to** David Manz (PNNL), Jozsef Gacsal (SecurityLit), Jason Carter (ORNL), Ralph Lange (ITER), Bob Dalesio, Michael Davidsaver (Osprey DCS), George McIntyre (Level-N Ltd)

**Contents**

1. Accelerator computing
2. Example Cyber Statistics, Regulations, and Thinking
3. Typical Cyber Computer Architecture for Accelerators
4. Conducting a Cyber Security Review
5. Extant Accelerator Control System Cyber Issue, EPICS
6. Improving EPICS cyber security
7. New Cyber Regulatory Framework, Compliance Challenge, and Future
8. Summary

**SLAC ACCELERATOR CYBER REVIEW FINDINGS**

**Positive Overall.** Our cyber security is complete with respect to common practice.

1. Login security is **comparable to most facilities**. Will soon be leading
2. **Backups are complete**
3. Malware Detection (CrowdStrike) & Vulnerability Detection are complete (subject to acceptance of the common principle that the control system be exempt from these).
4. CA Security (authorization to change PV) is designed, in cryo IOCS, and ready for broad implementation

**However, EPICS is insecure.** Its use is based on aging assumption of secure perimeter.

1. EPICS protocols lack strong authentication
  - a. Man in the Middle attack. A PV could be changed without ACR knowledge
  - b. EPICS users will be authorized for PV changes, but aren't presently strongly authenticated
2. IOC Software is not certificate authenticated (user can't be sure the IOC they're talking to is not an imposter)

**Additionally, some administration and management:**

1. PV drive limits are not all set – can lead to machine errors
2. Understaffed with Oracle DB Admin

Slide credit: Gregory White <https://indico.cern.ch/event/1270052/contributions/5598148/>

# Recommendations

## EPICS Controls Security Issues & Recommendations

SLAC

- **Passive Traffic Inspection**  
Passive attacker can observe and learn Process Variable and server names. Not considered serious.  
⇒ Could Mitigate by TCP+TLS(\*)
- **PV Denial of Service by search spam**  
Active attacker responds to PV search requests, directing to null server
- **PV Search Hijacking / Man in the Middle Attack**  
Active attacker responds quickly to all observed searches, **redirect clients to rogue EPICS server.**  
Returns fake data, or proxy forwards bad control data to a legitimate control system EPICS server.  
Very bad things.  
⇒ **Mitigate by adding Transport Layer Security** (as long as attacker does not hold certificate)
- **Server impersonation / credential theft**  
Theft of server certificate used to maliciously impersonate PVs provided a legitimate server.  
⇒ **Mitigate by something like certificate “pinning.”**

(\*) Transport Layer Security (TLS); Encryption, certificate-based authentication, compression.

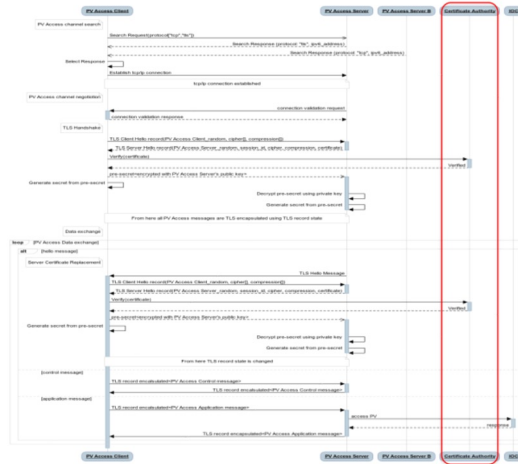
*Work of Michael Davidsaver, Osprey DCS, under SLAC contract, 2021*

*Slide credit: Gregory White <https://indico.cern.ch/event/1270052/contributions/5598148/>*

# EPICS changes

## EPICS Security Improvement: PvAccess and Transport Layer Security (TLS)

SLAC



- PvAccess += Transport Layer Security.
- First step to Zero Trust Architecture in EPICS Accelerator controls
- Multi phase project:
  - Server side authentication
  - Client side
  - Yr 2 / 3. Certificate server? Name Server? Pinning?
- Transition phase: EPICS **pvAccess TLS fully backward compatible**. TLS endpoints co-exist with non TLS.
- GOAL: All endpoints use PvAccess + TLS.
- NOTE security implies removal of Channel Access protocol from EPICS systems (!)
- Very early Prototypes now available.

Figure: EPICS PVA negotiation with TLS proposal, showing: TLS handshake after message validation, "tls" message, cipher handshake, and certificate verification additions to EPICS PvAccess protocol. Modification uses the "magic" byte in the pvAccess header, and existing protocols field in the search response.

Work of George McIntyre, Level-N Ltd, + Michael Davidsaver, Bob Dalesio, Osprey DCS, for SLAC. Kay Kasemir for ORNL.

## EPICS Security Improvement 2:

### US Dept of Energy sponsored project to add TLS to PvAccess for the EPICS community

\$ 1.4 M over 2 years. SLAC leading, Osprey primary contractor (M. Davidsaver, G. McIntyre). Kay Kasemir (ORNL) adding for core-pva. Many, many thanks to Dale Dale Hugo Leschnitzer & Mark Hahnert (US Dept of Energy, Office of Science).

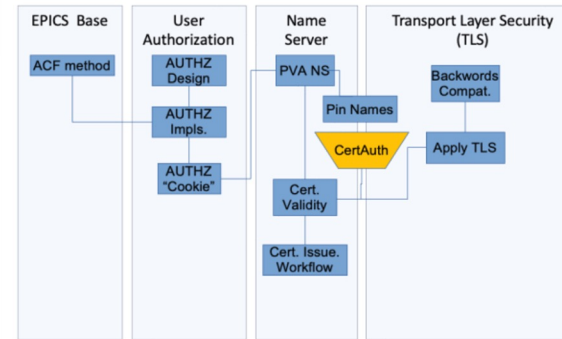


Figure: The basic architecture of PVA+TLS in context of an EPICS installation, showing integration with enterprise authentication, certificate validation and management, and integration with EPICS Access Control.

#### Year One

1. Technology analysis and selection
2. Implement a simple prototype containing TLS additions to the EPICS PV Access protocol
3. Design changes to EPICS
4. Implement prototype for PV Access Name Server changes
5. Implement prototype for client authorization via certificate
6. Integrate with site-specific authentication protocols
7. PVA Protocol Performance tests
8. PV Access Name Server Performance tests
9. Functional Verification Testing and Combined Report

#### Year Two

1. Implement Beta version of PVA TLS
2. Update Beta with performance and usability improvements
3. Update Beta with certificate management improvements -
4. Release Beta to SLAC and monitor
6. Release Beta to selected facilities and monitor results

Slide credit: Gregory White <https://indico.cern.ch/event/1270052/contributions/5598148/>

# Considerations for PVA over TLS

*PVA: process variable access  
(like a Tango attribute)*



## What needs protection?

<b>Yes</b>	<b>No</b>
• Unauthorized PUT	• Secrecy
• Tampering with GET/MONITOR <i>Trick authorized user into Making incorrect PUT</i>	<i>May come incidentally, just not required</i>

---


## Threat model

- Actors
  - Passive attacker on adjacent host *Same subnet*
  - Active attacker on adjacent host *Same subnet*
  - Attacker on client host *Same host*
  - Attacker on server host *Same host*
  - Compromised client *Same process*
  - Compromised server *Same process*



## Threat Vectors (2)

• Passive traffic inspection (TCP/UDP)	Reasonable © Mitigation? TLS/??? <i>Doesn't matter</i>
• Denial of service by search spam	NameServer
• Search hijacking	NS
• Server impersonation	TLS
• Server credential theft	NS + cert. pinning



Slide credit: Michael Davidsaver, George McIntyre <https://indico.fnal.gov/event/58280/contributions/264558/>



# Considerations for PVA over TLS

NTP: Network Time Protocol

## System Considerations

- Distributing CA certs.
  - Straight forward copying of (mostly) static files
- Issuing Server (and Client) certs.
  - Tedious ~manual process
  - What Common Name?
- Cert. validity
  - Expiration date?
  - Certificate Revocation List?
  - **Periodic online check** (Open Certificate Status Protocol)?



## Certificate Validity

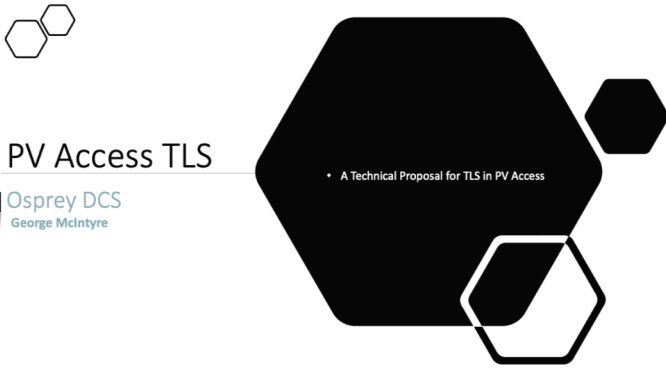
- Time based
    - Valid between X and Y
    - Encoded in certificate
  - CRL
    - Periodically published list of revoked (bad) certs.
  - ~~Open Certificate Status Protocol~~ Online Certificate Status Protocol
    - Access to database of signatures on valid certificates (w/ time)
    - Like an expiration date
    - Can be updated **w/o reissuing** cert.
    - Requires client connection to OCSP server(s)
- "stapling" helps*

*Trust in NTP becomes critical*



Slide credit: Michael Davidsaver, George McIntyre <https://indico.fnal.gov/event/58280/contributions/264558/>

# Detailed technical proposal for EPICS TLS



PV Access TLS

Osprey DCS  
George McIntyre

• A Technical Proposal for TLS in PV Access

EPICS Collaboration Meeting, Fermilab, 2023

## Current Work – commissioned by SLAC - 2023

**PV Access TLS Feature Implementation:**  
*George McIntyre*

- TLS Channel Search over TCP
- TLS Handshake
- TLS Encapsulation with Encryption and Signature
- Support for Server Certificate & Rotation
- Support for Compression
- Command line tool support – *pvput, pvget, pvmonitor, ...*
- Unit test suite

**EPICS Technical Security Analysis:**  
*Michael Davidsaver*

- Report
- Roadmap

**Java implementation** – *maintainer Kay Kasemir*

- <https://github.com/ControlSystemStudio/phoebus/tree/master/core/pva>

**C++ implementation** – *maintainer Michael Davidsaver*

- <https://mdavidsaver.github.io/pvxs>

**Documentation**

- <https://github.com/epics-base/pvAccessCPP/v>

- **Features**
  - Server Authentication
  - Encryption
  - Server Certificate rotation
  - Compression
  - Client Authentication
  - Authorization
- **How it works?**
  - The low level details

## Out of scope

**Features**

- Client Certificates
- Client configuration mappings for TLS parameters
- Add TLS to Channel Access
- UDP Broadcast search
- UDP response
- Beacon messages
- Add additional TLS beacon messages for servers supporting both TLS and TCP
- Any changes to support TLS in Gateways
- Any changes to support TLS in EPICS Python (pvaPy)
- Any changes to support TLS in PV Database

**Repositories**

**EPICS base Java**

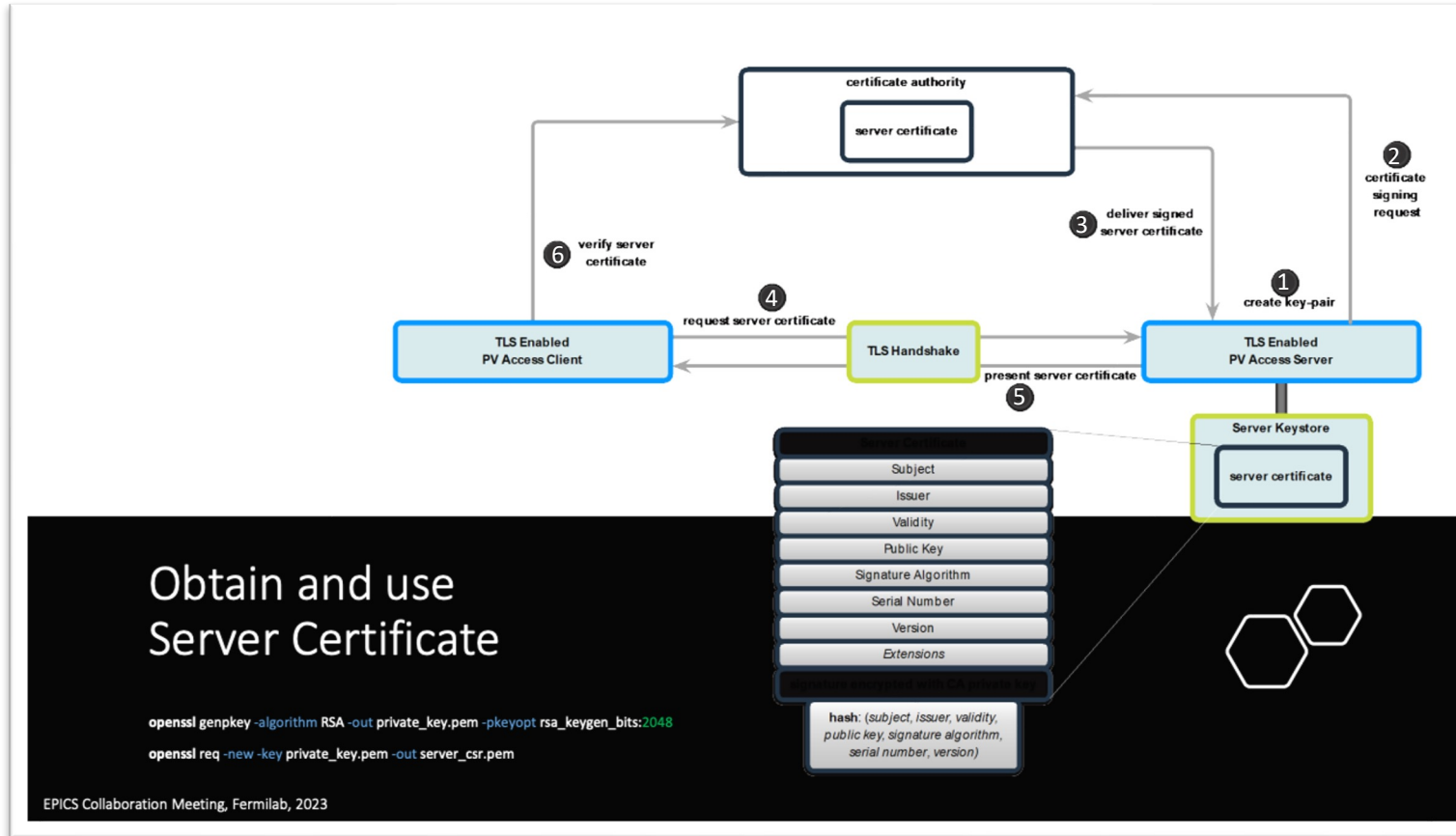
- <https://github.com>
- <https://github.com>

**EPICS base C++**

- <https://github.com>
- <https://github.com>

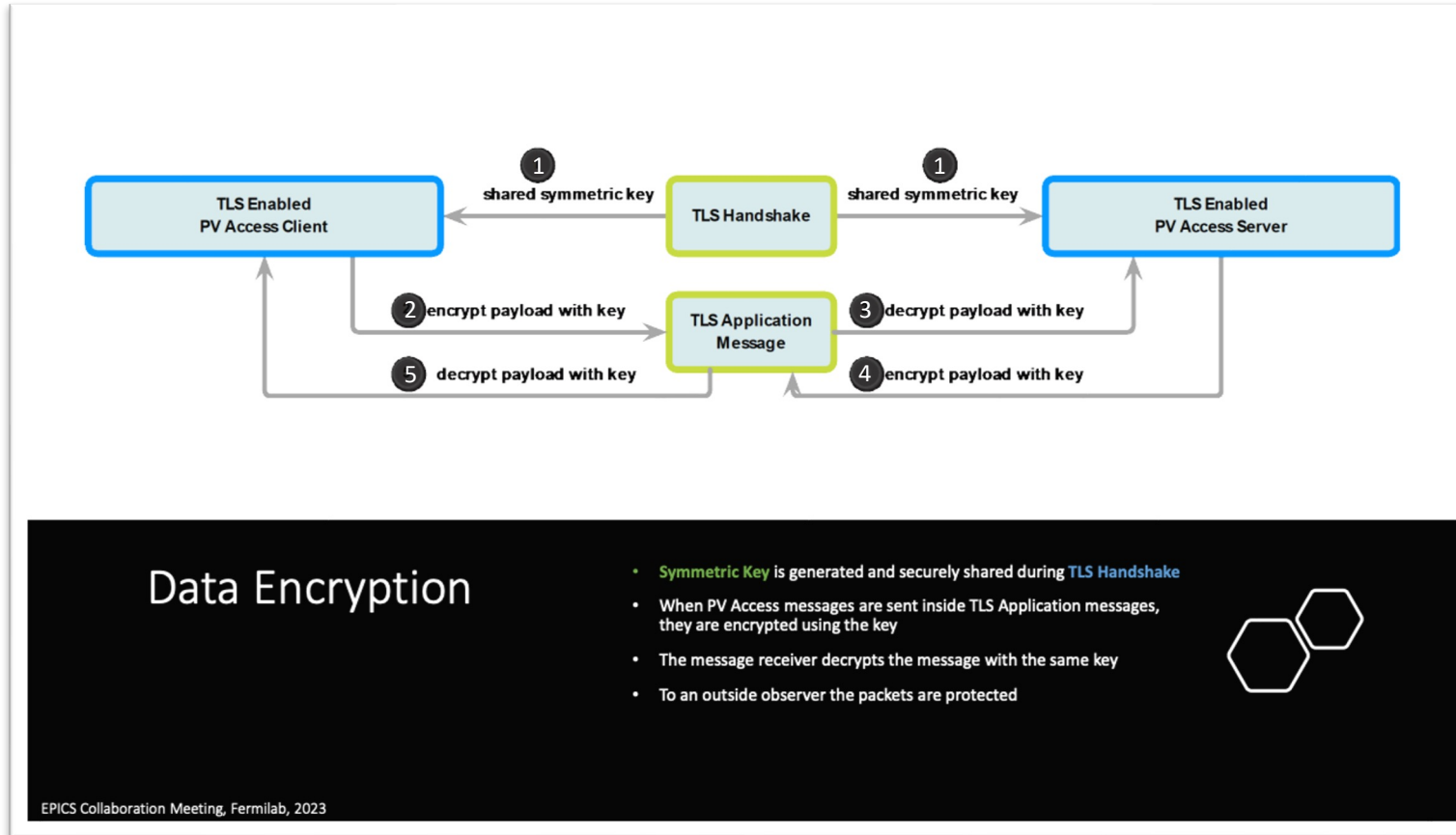
Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>

# Certificate usage



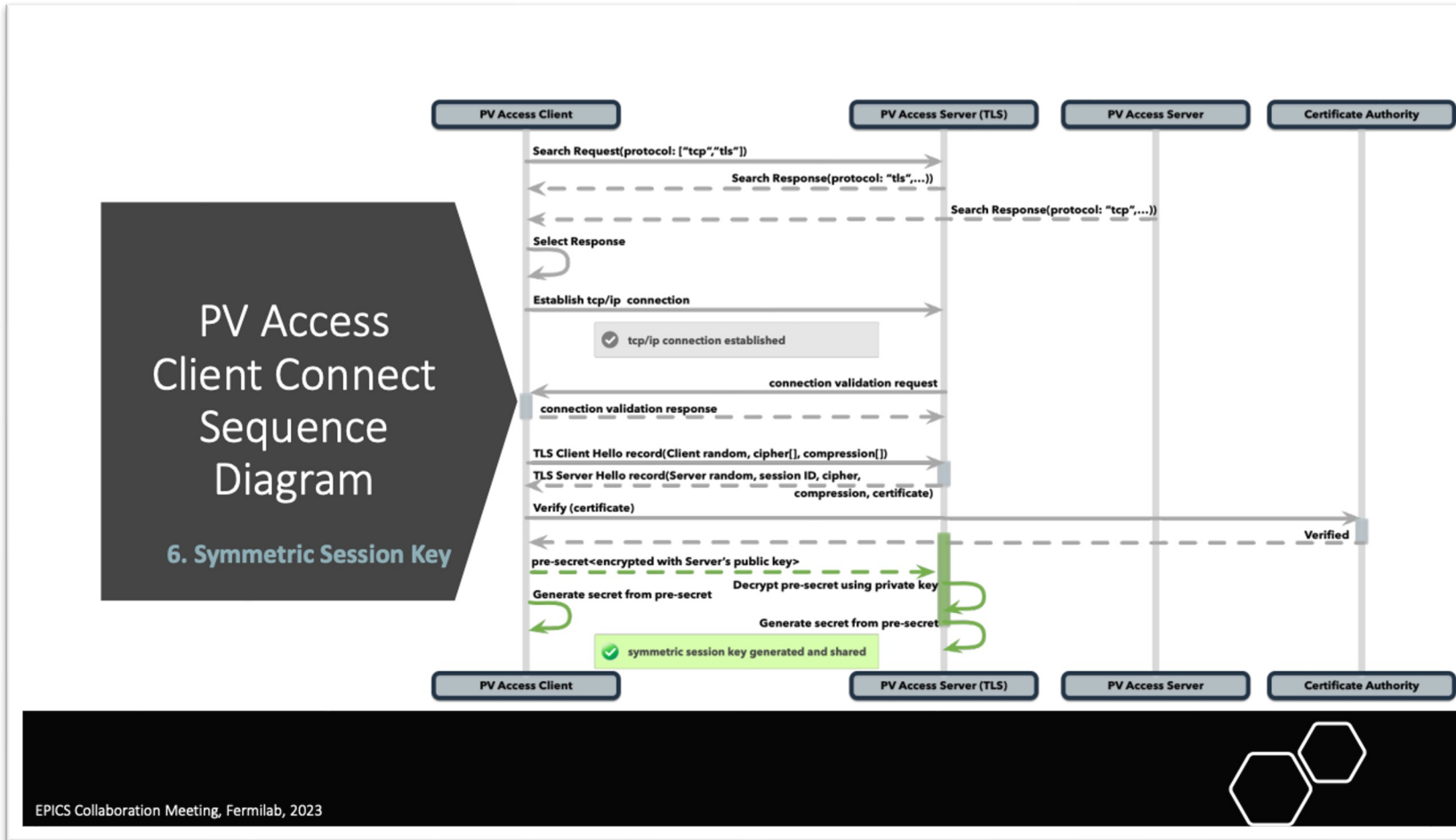
Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>

# Data encryption



Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>

# Establishing TLS session



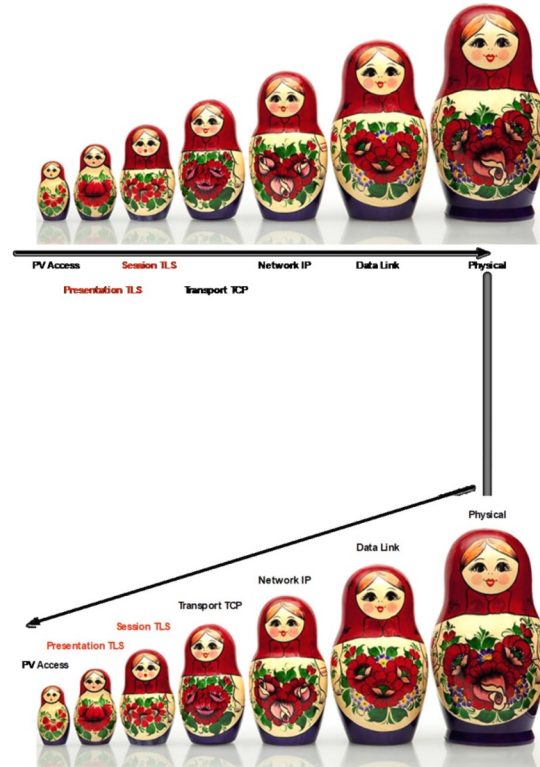
Requires about 6.5kB of data!

<http://netsekure.org/2010/03/tls-overhead/>

Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>

# Network stack

Encapsulation  
of PV Access  
Messages  
+ TLS



Overhead ~40 bytes  
per message

<http://netsekure.org/2010/03/tls-overhead/>

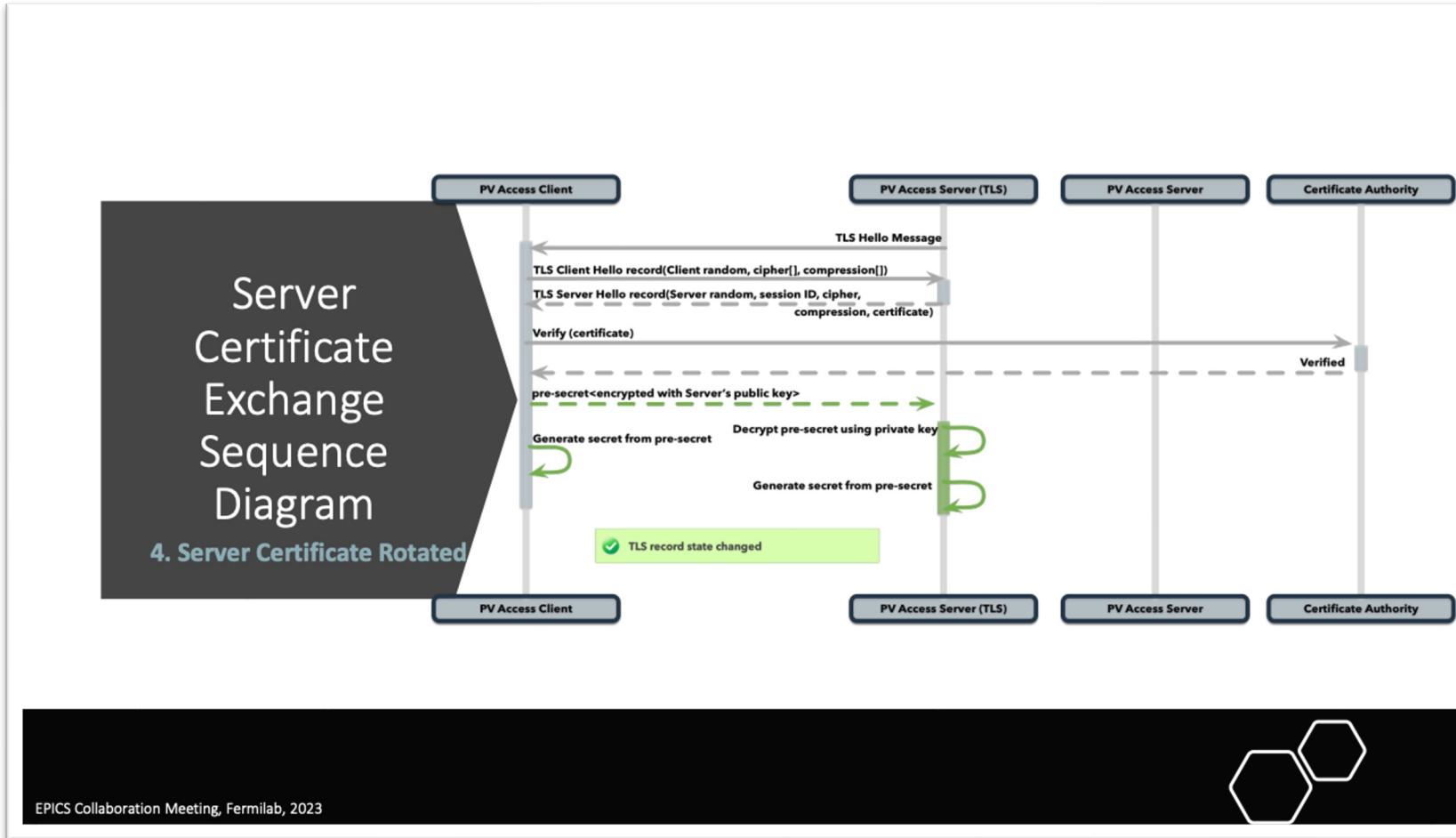
EPICS Collaboration Meeting, Fermilab, 2023

Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>

# Server certificate rotation

Can be done while client remains connected

Does client drop connection after certificate validity expires?



Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>



# Outcome

## What will adding TLS get us?

### TLS Benefits

- **Server Certificates** →
  - Prevent **Service Impersonation**
  - Prevent **Man-in-the-Middle** attacks
- Cipher suite **Message Authentication Codes** →
  - Guarantee **Data Integrity**
- Securely shared **Symmetric Session Keys** →
  - Prevent **Packet Snooping**
- **Client Certificates** →
  - Provide a mechanism for **Service Access Control**
  - **Protect Data** by allowing Services to Restrict Access
  - Can be used as part of strategy to **Reduce impact of DoS Attacks**

### TLS Will Not

- Prevent **PV Impersonation** in a mixed TLS/TCP network
- Prevent discovery of **Service Endpoint** or **PV name**
- Prevent discovery of **Encryption Type**
- Prevent discovery of **Data Transmission Frequency**
- Prevent discovery of approximate **Amount of data transmitted**



### Network Management Impact

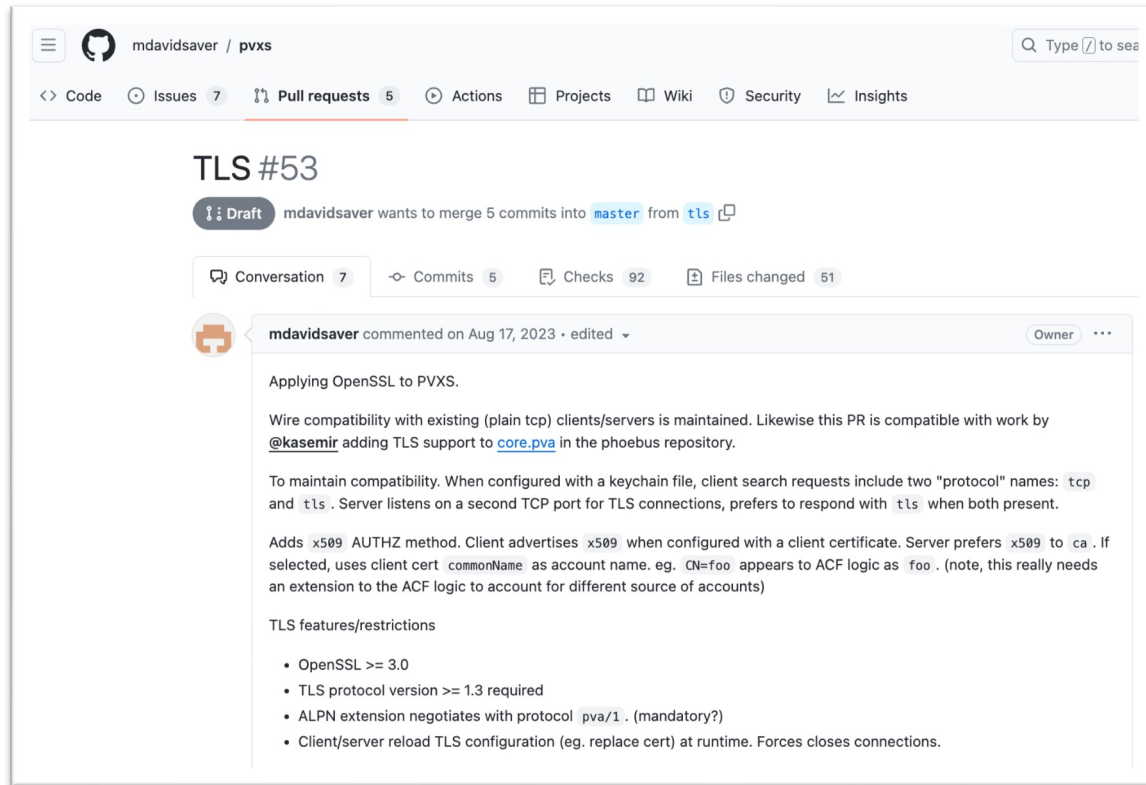
Install Server Certificates  
Configure Network for TLS traffic  
(network management tools)

EPICS Collaboration Meeting, Fermilab, 2023



Slide credit: George McIntyre <https://indico.fnal.gov/event/58280/contributions/265752/>

# EPICS: C++ changes for PV access over TLS



The screenshot shows a GitHub pull request titled "TLS #53" for the repository "mdaidsaver / pvxs". The pull request is in a "Draft" state and aims to merge 5 commits into the "master" branch from the "tls" branch. The interface includes navigation tabs for Code, Issues (7), Pull requests (5), Actions, Projects, Wiki, Security, and Insights. Below the title, there are statistics for the pull request: Conversation (7), Commits (5), Checks (92), and Files changed (51). A comment by the user "mdaidsaver" is visible, dated August 17, 2023, and marked as the owner. The comment text is as follows:

Applying OpenSSL to PVXS.

Wire compatibility with existing (plain tcp) clients/servers is maintained. Likewise this PR is compatible with work by @kasemir adding TLS support to [core.pva](#) in the phoebus repository.

To maintain compatibility. When configured with a keychain file, client search requests include two "protocol" names: tcp and tls. Server listens on a second TCP port for TLS connections, prefers to respond with tls when both present.

Adds x509 AUTHZ method. Client advertises x509 when configured with a client certificate. Server prefers x509 to ca. If selected, uses client cert commonName as account name. eg. CN=foo appears to ACF logic as foo. (note, this really needs an extension to the ACF logic to account for different source of accounts)

TLS features/restrictions

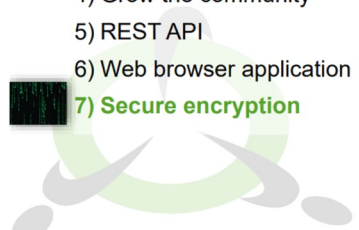
- OpenSSL >= 3.0
- TLS protocol version >= 1.3 required
- ALPN extension negotiates with protocol pva/1. (mandatory?)
- Client/server reload TLS configuration (eg. replace cert) at runtime. Forces closes connections.

<https://github.com/mdaidsaver/pvxs/pull/53>

# What about Tango?

# Tango roadmap 2015

From ICALEPCS 2017



**ROADMAP**

Roadmap from ICALEPCS 2015 (as described in WEA3001):

- 1) Improve documentation
- 2) Move to Git
- 3) Remove CORBA completely
- 4) Grow the community
- 5) REST API
- 6) Web browser application
- 7) Secure encryption**
- 8) Database performance
- 9) Device class Marketplace
- 10) Long Term Support
- 11) Tango Virtual Machine
- 12) Auto-Generate Unit tests
- 13) SysML support
- 14) Replace Boost.Python

26 TANGO Kernel Status - ICALEPCS 2017 - Barcelona



**TANGO**

---

**ROADMAP #7: SECURE ENCRYPTION**

Implement a secure encrypted protocol for public networks

- **Security managed by infrastructure**
  - VPN
  - Web server
  - HAProxy



27 TANGO Kernel Status - ICALEPCS 2017 - Barcelona

**TANGO**

Slide credit: Reynald Bourtembourg [https://accelconf.web.cern.ch/icalepcs2017/talks/mobpl02\\_talk.pdf](https://accelconf.web.cern.ch/icalepcs2017/talks/mobpl02_talk.pdf)

# Tango protocols

Remote Procedure Calls use CORBA

Commands, reading/writing attributes

C++ uses omniORB – it has TLS option (TLS 1.3 support?)

Java use JacORB – provides IIOP over SSL

Events use zeromq

libzmq has:

no TLS

CurveZMQ for encryption: <https://rfc.zeromq.org/spec/26/>

ZAP for authentication: <https://rfc.zeromq.org/spec/27/>

Events are read-only, so do we need authentication?

# Common issues



# Common issues

## Certificates

Certificate Authority

How do servers get certificates?

How do clients get certificates?

Expiry, revocation, rotation

If time-based, then NTP must be secure

## Backwards compatibility

Do we allow insecure clients/servers?

If yes, will we ever turn off the insecure option?

# Common issues

## Performance

- Time to establish connection

- Latency

- CPU/memory requirements

- Network bandwidth (~6.5kB for TLS session, ~40 B extra per message \*)

## General

- Is CA a single point of failure?

- Can we get locked out of our own system?

- How to sign software applications?

- Authentication?

- Authorisation?

- Accounting?

- Secrets management

\* TLS overhead for TLS 1.2: <http://netsekure.org/2010/03/tls-overhead/>

# Use cases

# Use cases

## Tango servers

Launched by Starter service / Kubernetes

Started manually by engineer

## Client applications

Jive, Sardana, Taurus, Taranta

User scripts

Client to device in same process – does it do TLS?

# Use cases

Tango device/client developers

Unit tests run locally – do we want to deal with certificates?

CI/CD

Security team

Penetration testing

Auditing installed software

Want vulnerable libraries updated ASAP

”Break-glass” procedure

Procedure to report and handle vulnerabilities

# Use cases

Updating server and client certificates

- Initial deployment

- Rotation/update after expiry

- Revocation – how soon is connection dropped?

Updating cipher suite and cryptographic keys

- security update to library

- need more bits

- need new cipher suite

- old & new versions of Tango, with different encryption methods



# Use cases

## Authentication

Can we use LDAP, SSO?

Can we use WebAuthn, hardware keys/devices?

Are certificates linked to user accounts, hosts, or device servers?

How to handle beamline user accounts?

How to handle service accounts?

How to handle temporary accounts for visiting users?

# Use cases

## Authorisation

Is Tango access control sufficient?

Are the rules from TAC sufficient?

How fine-grained do we need it?

Something like EPICS access control list files with rules?

OAuth2?

# Use cases

## Accounting / Auditing

What do we log?

Where do we log?

Who has permission to see the logs?

Who has permission to change/delete?

Is there some notification for suspicious activity?

# Conclusion

# Conclusion

Re-use existing standards and technologies

Learn from other control systems

It will take a long time / a lot of resources

Encrypting Tango data is only a small part of cyber security!

MAXIV