Secure authentication mechanisms

Mikołaj Dobski, Paweł Węgrzak
WP9T2 / PSNC

PRACE Days 2019, Poznań, 2019-05-15

www.geant.org

Agenda

- brief SymbIoTe H2020 project introduction
- Multi-level security
  - Cohesive design driven responsibility separation
- Workshop part 1. Mutual Authentication
- Workshop part 2. Attributes Based Authorization Control

Goals (Learn about):

- security best practices & design goals **YOU** can (should) apply
  - and tools to support them
  - any why it is safer not to reinvent the wheel
- our design mistakes
  - and how **YOU** can avoid them
- why some security related questions cannot be (simply) answered

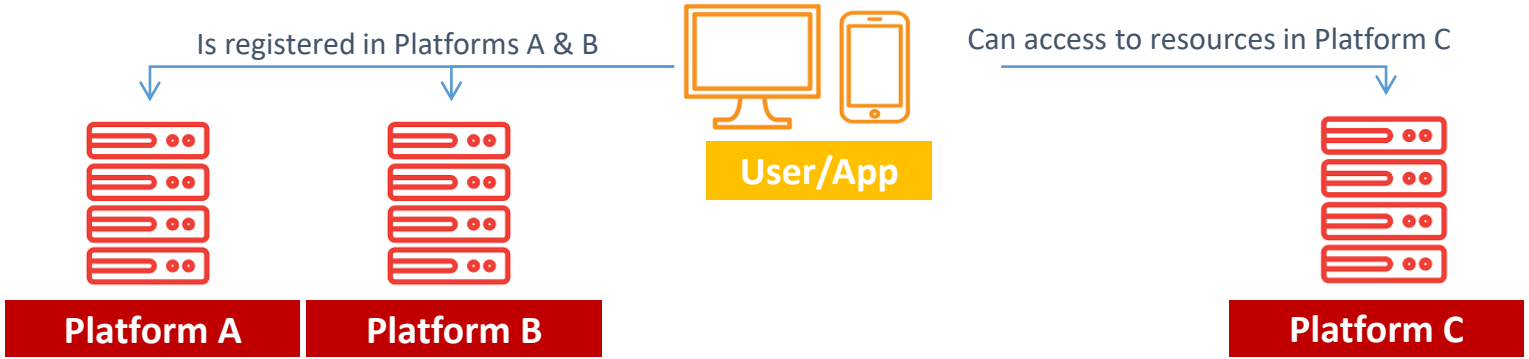
H2020 SymbIoTe Motivation: A simple interoperable IoT app

- Universal light switch on your mobile phone
  - … switch on/off the lights wherever you go (at home, in the office, in public spaces…)
  - … but of course, only if you are allowed to do so…

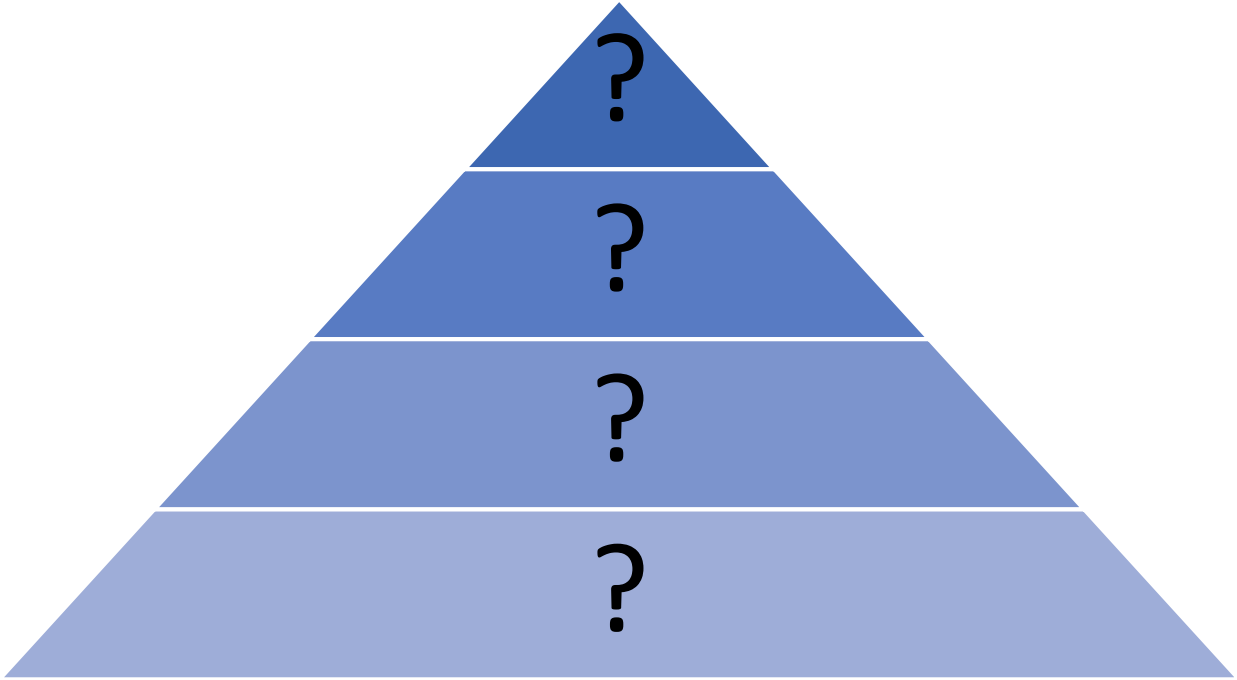
Main goal and approach

- Target goals:
  - **generic…**
  - **secure…**
  - **multi-domain…**
  - **attributes-based**
  - **access right composition.**

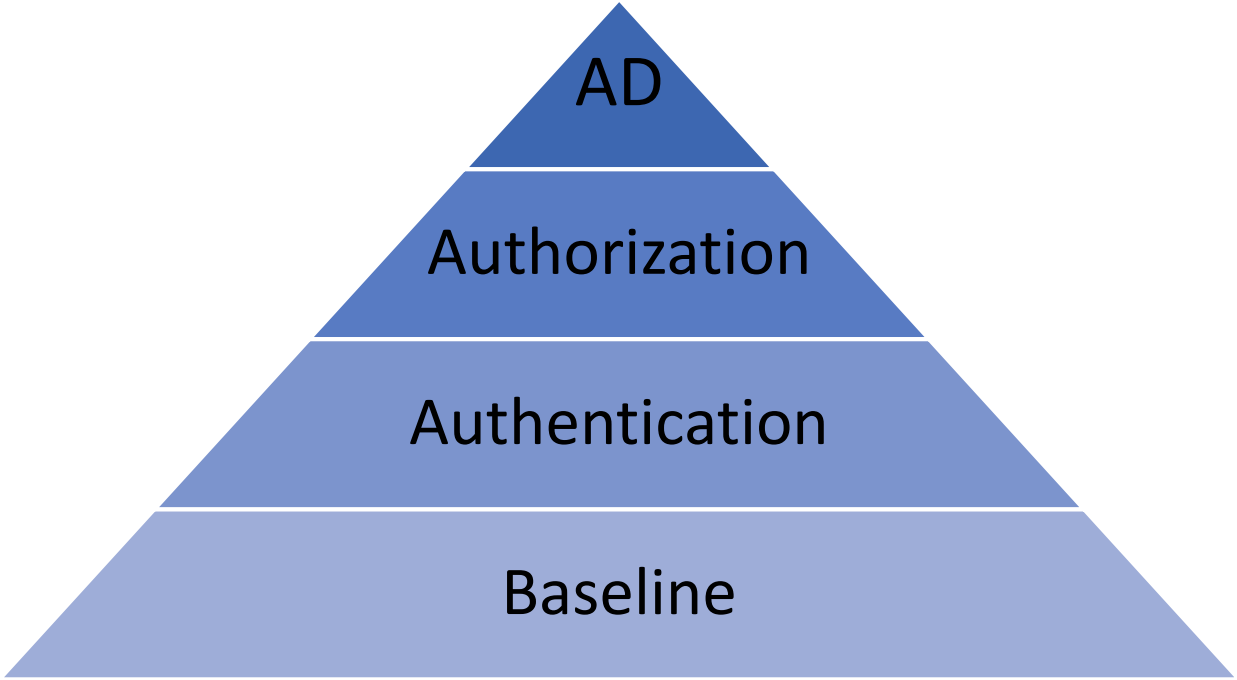
Authorizing users registered in one or more A&A authorities to access resources exposed elsewhere



Security solution layers (0)

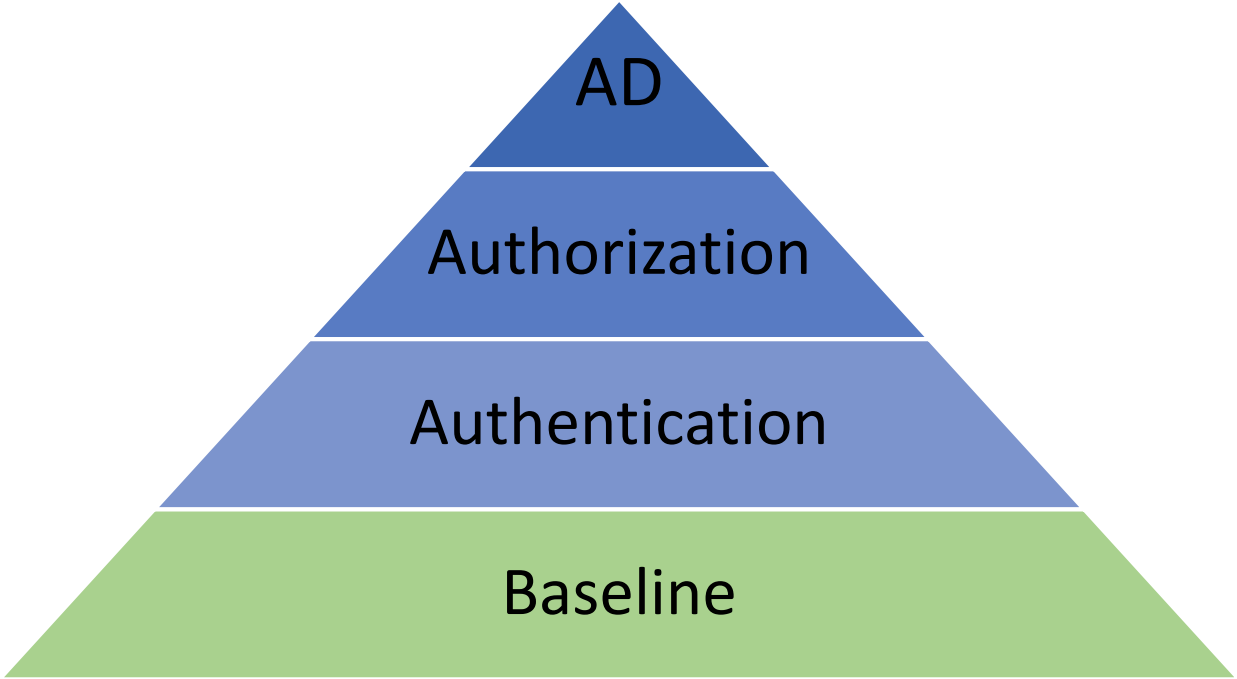


Security solution layers (0)

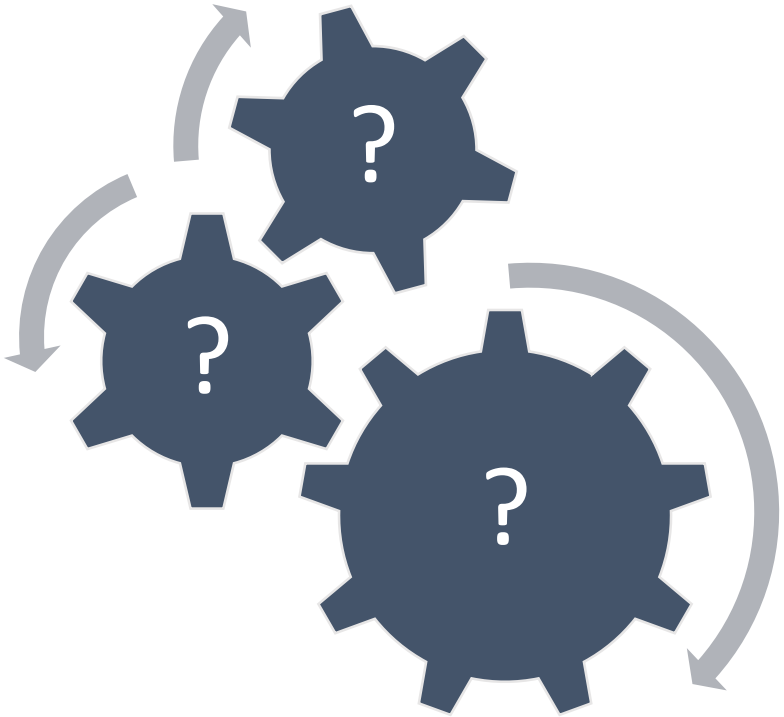


Divide (your design) and conquer!

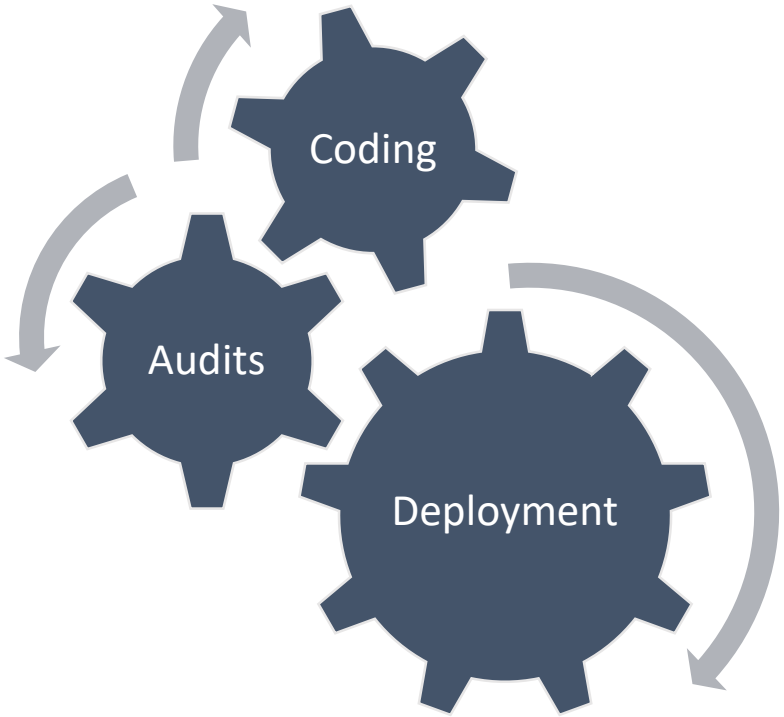
Security solution layers (1)



Baseline security



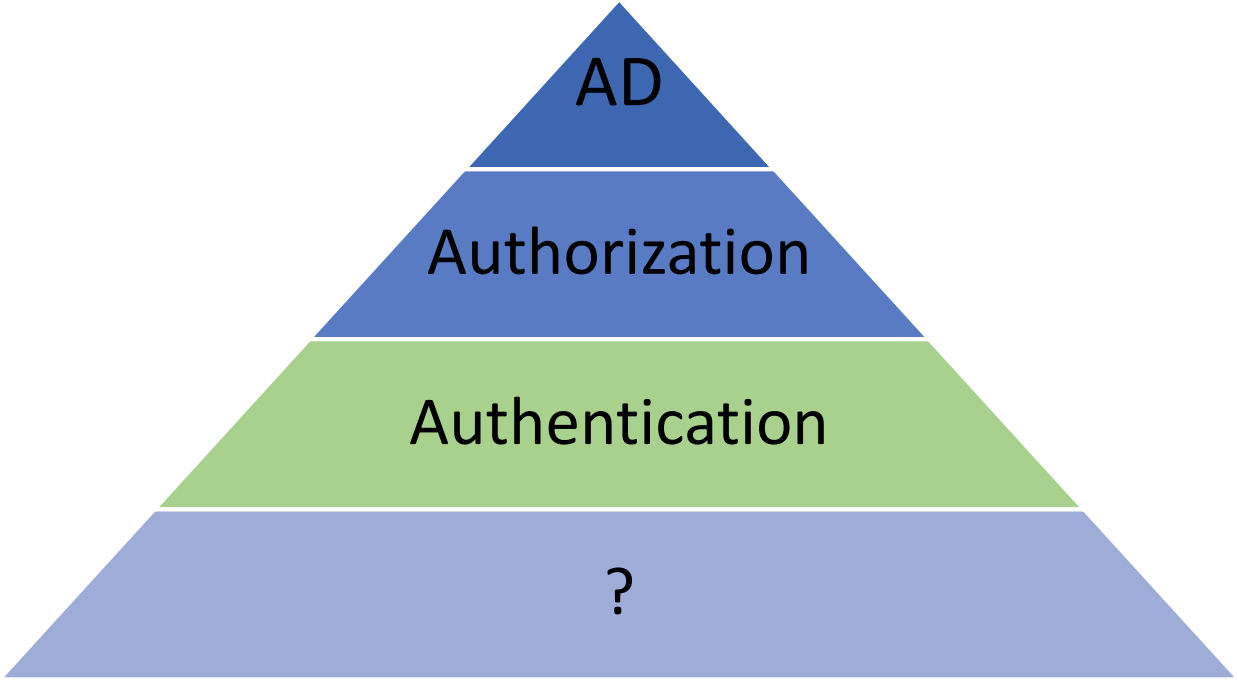
Baseline security



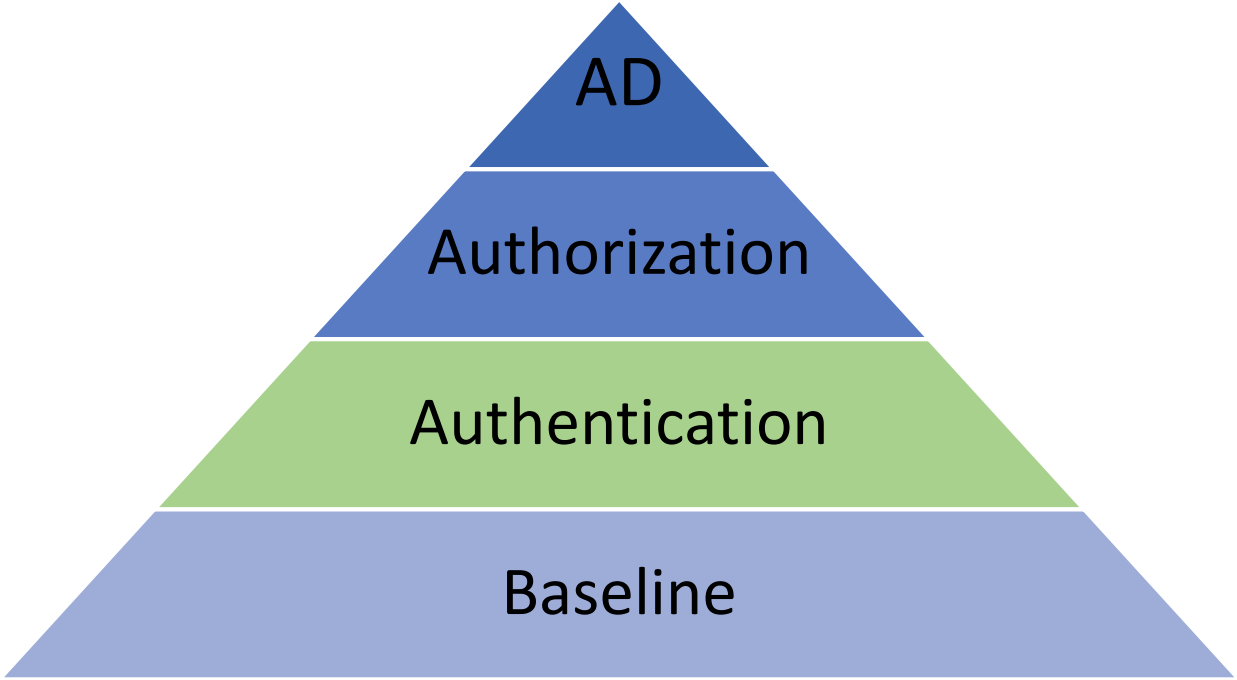
Ask SecOps for assistance in baseline security

Code quality ahead of features

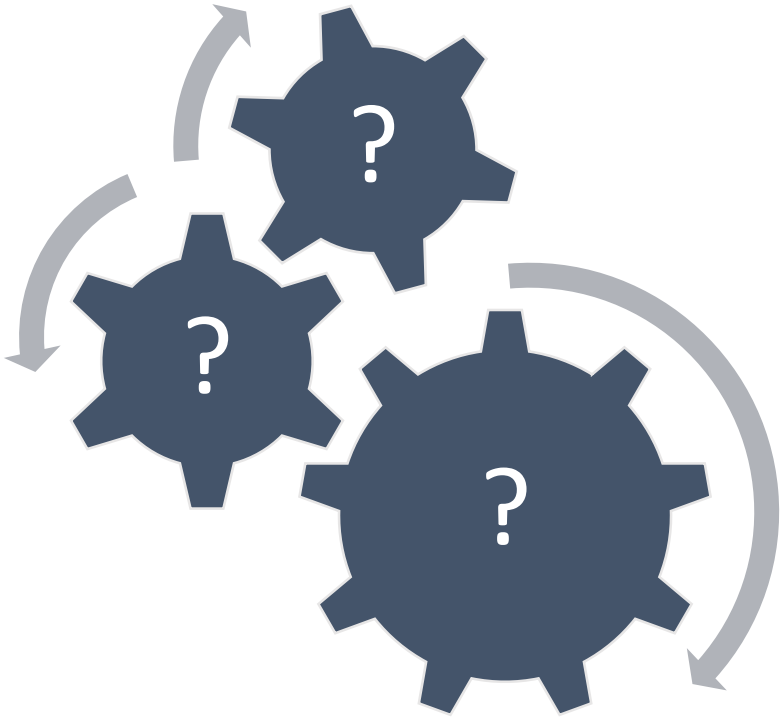
Security solution layers (2)



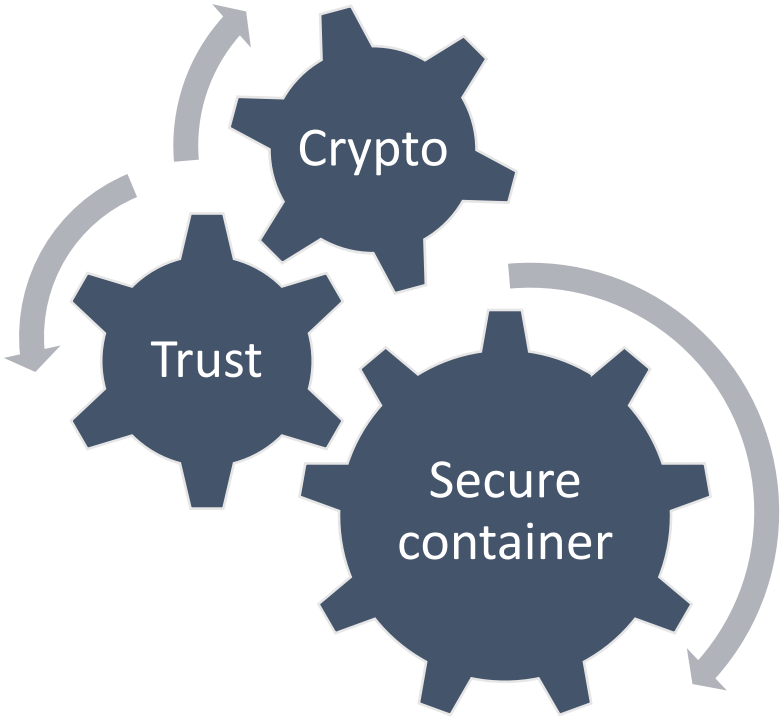
Security solution layers (2)



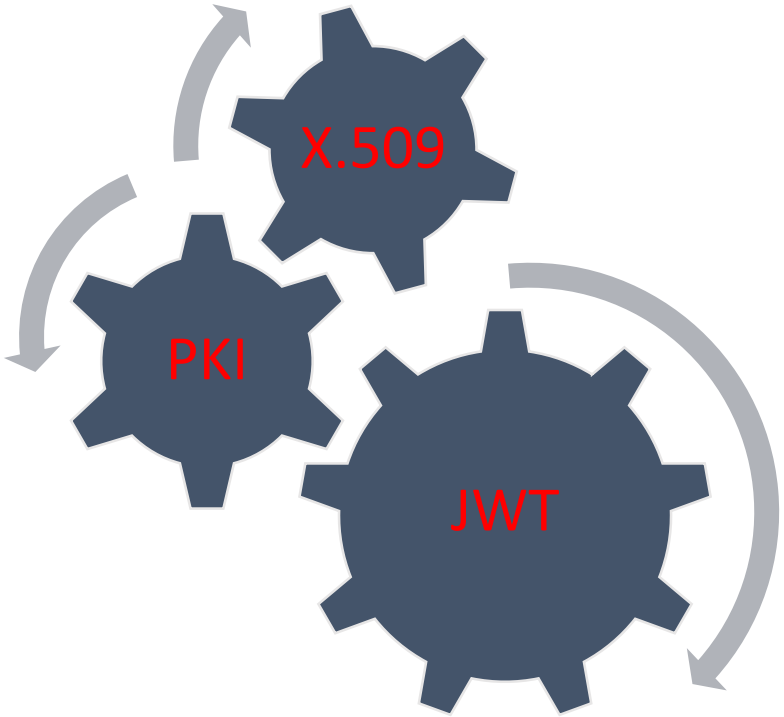
Authentication layer components (concepts)



Authentication layer components (concepts)



Authentication layer in SymbIoTe



Security material containers - JSON Web Tokens

- Well-known structure* used for storing client's attributes
  - Basically a key-value map
- Extendable with any custom claims (attributes)
- Highly configurable as can be used to provide
  - Trust -> JWSignature or
  - Confidentiality -> JWEncryption

*RFC 7519 https://tools.ietf.org/html/rfc7519
resources https://jwt.io/

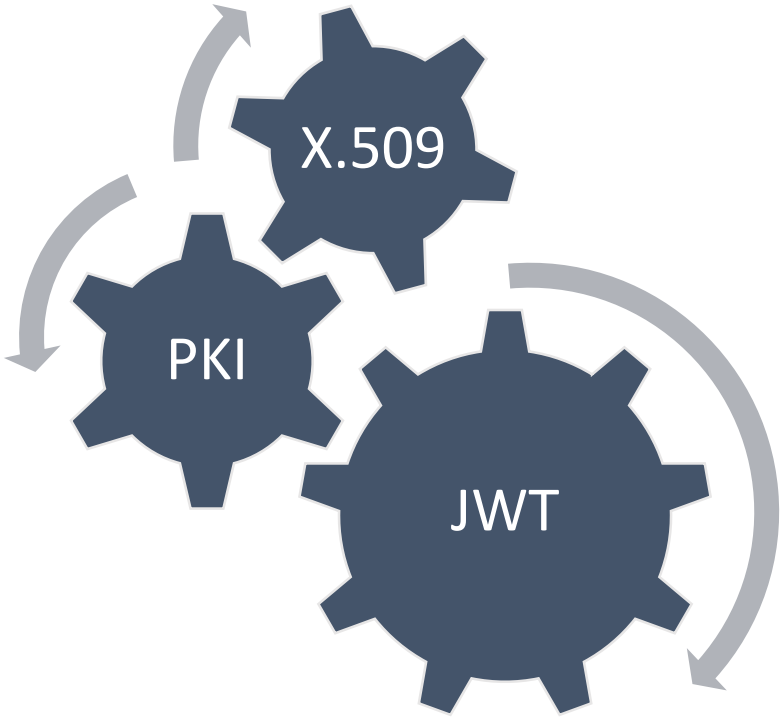
Security material containers - JSON Web Tokens

- Well-known structure* used for storing client's attributes
  - Basically a key-value map
- Extendable with any custom claims (attributes)
- Highly configurable as can be used to provide
  - Trust -> JWSignature or
  - Confidentiality -> JWEncryption

jti = JWT_ID
alg = $ECDSA_{256}$
iss = AAM_ID
sub = APP_ID
iat = ISSUE_DATE
exp = EXPIRATION_DATE
ipk = AAM_PUBLIC_KEY
spk = APP_PUBLIC_KEY
att = ATTRIBUTE_VALUE
ttyp = TOKEN_TYPE
sign = SIGN-$ECDSA_{256}$($H(T_U)$, AAM_PRIVATE_KEY)

*RFC 7519 https://tools.ietf.org/html/rfc7519
resources https://jwt.io/

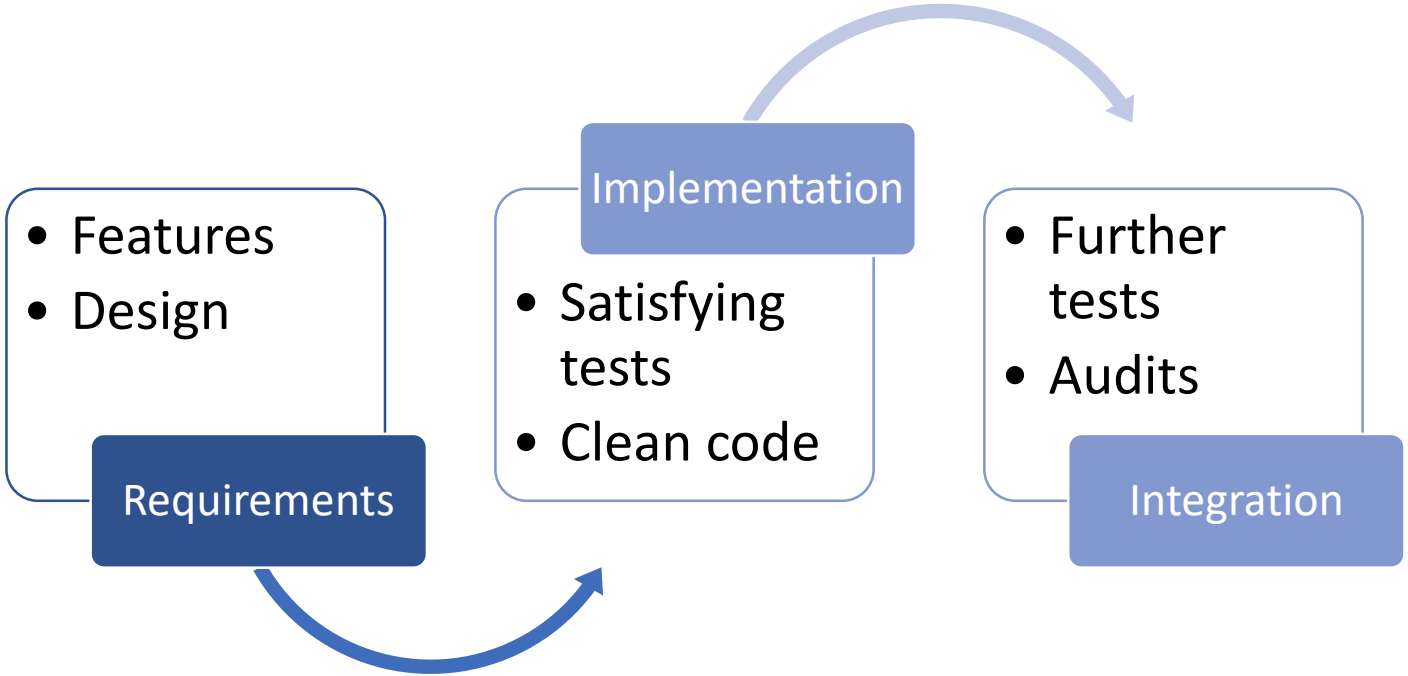
Authentication layer



Short relief

Let’s talk about SDLC

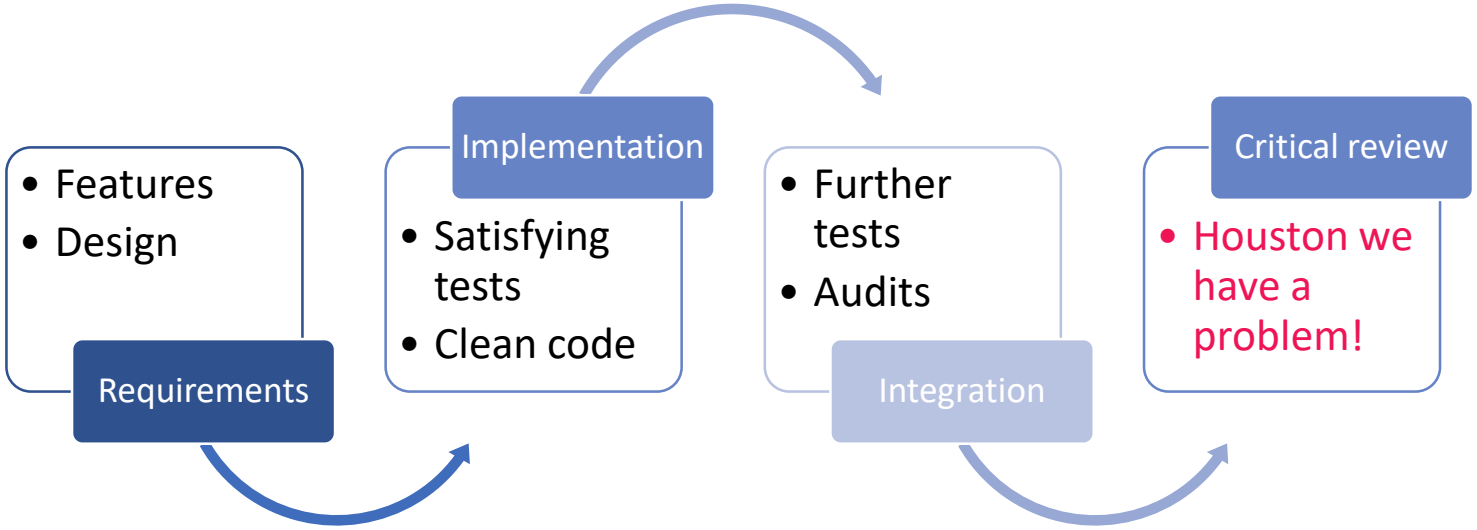
Software development life-cycle



Initial implementation (release) review results

- PKI
  - actors' management done
  - issuing & revoking certificates by username and password done
- JWT
  - Generating Auth(Z) payloads for actors
    by Auth(N) using username and password done
- Authorization using acquired JWT is working
- Ready to code new features! :D

Software development life-cycle



Initial implementation (release) review results

- PKI
  - actors' management done
  - issuing & revoking certificates by username and password done
- JWT
  - Generating Auth(Z) payloads for actors by Auth(N) using username and password done
- Authorization using acquired JWT is working
- Ready to code new features! :D

Initial implementation (release) critical review results

- PKI
  - actors' management done
  - issuing & revoking certificates by username and password done
- JWT
  - Generating Auth(Z) payloads for actors
    by Auth(N) using username and password done
- Authorization using acquired JWT is working
- Ready to code new features! :D

Task 1

- Authenticate yourself and claim your certificate!

Security material containers - JSON Web Tokens

- Well-known structure used for storing client's attributes
- Extendable with any custom claims (attributes)
- Configurable for a variety of purposes
  - Authorization JWS – described later
  - Introduced
    - (authorization token) Acquisition JWS - two step process
      - First issuing certificate using username and master password
      - Daily basis use of the client certificate
    - …

Auth(Z) token Acquisition JWS

- Mandatory information
  - ?
  - ?
  - ?

Auth(Z) token Acquisition JWS

- Mandatory information
  - ?
  - ?
  - ?

alg = $ECDSA_{256}$
iss = ACTOR_ID
sub = CLIENT_ID
iat = ISSUE_DATE
exp = EXPIRATION_DATE
sign = SIGN-ECDSA256(H(T_U), A_PRIVATE_KEY)

Task 2

- Claim your token!

Security material containers - JSON Web Tokens

- Well-known structure used for storing client's attributes
- Extendable with any custom claims (attributes)
- Configurable for a variety of purposes
  - Authorization JWS – described later
  - (authorization token) Acquisition JWS - two step process
    - First issuing certificate using username and master password
    - Daily basis use of the client certificate
  - Other?

Security material containers - JSON Web Tokens

- Well-known structure used for storing client's attributes
- Extendable with any custom claims (attributes)
- Configurable for a variety of purposes
  - Authorization JWS – described later
  - (authorization token) Acquisition JWS - two step process
    - First issuing certificate using username and master password
    - Daily basis use of the client certificate
  - Authentication (challenge) JWS

Auth(N) challenge-response JWS

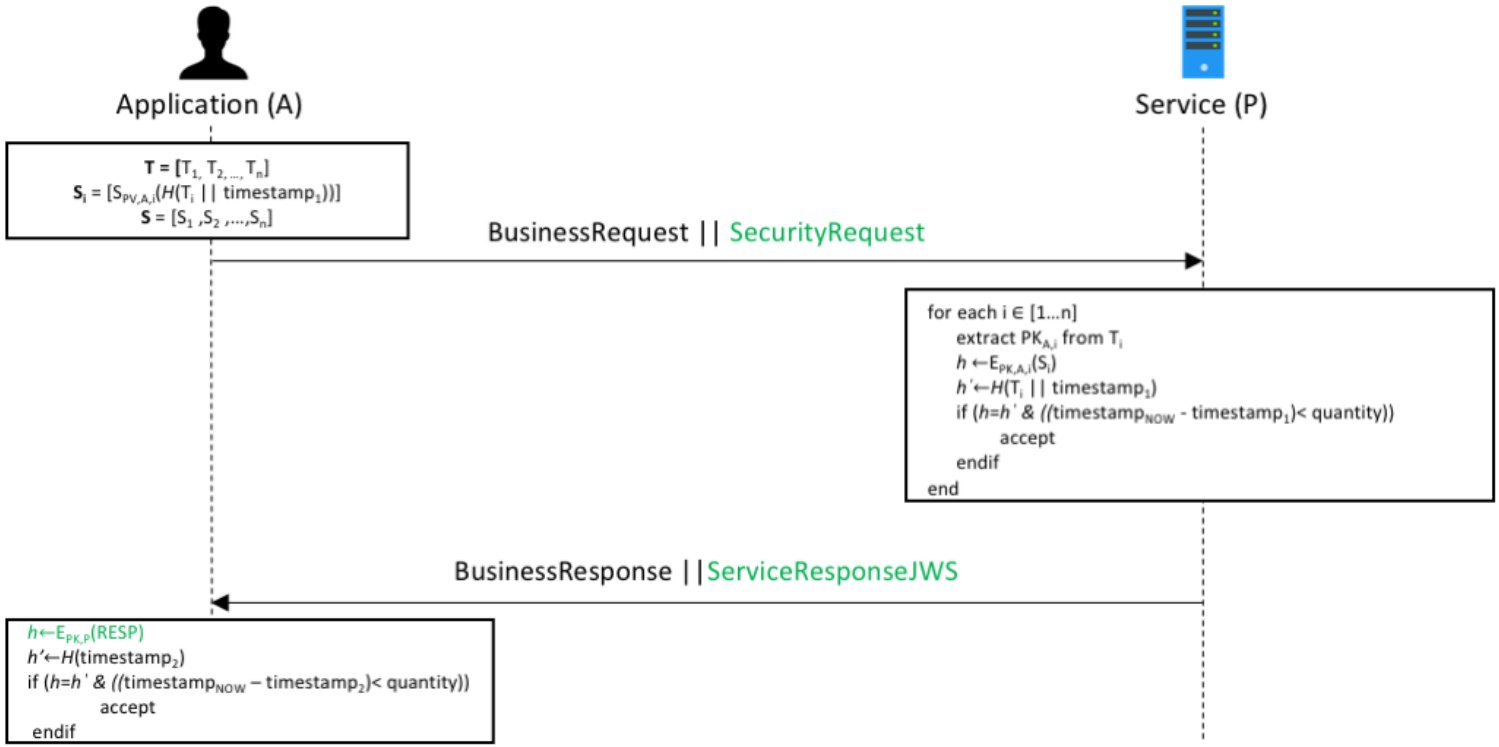
- Why?
- What to prove?
- How?

Auth(N) challenge-response JWS

- Why?
  - Auth(Z) token stateless
- What to prove?
  - Ownership / identity
- How?
  - By signing
  - Including timestamp!!!

iss = sub$_T$ = APP_ID
sub = jti$_T$ = JWT_ID
ipk = spk$_T$ = APP_PUBLIC_KEY
hash = hashed(TOKEN+TIMESTAMP)
iat = ISSUE_DATE
exp = EXPIRATION_DATE
sign = SIGN-ECDSA$_{256}$(H(T_U), APP_PRIVATE_KEY)

Auth(N) by challenge-response JWSes



Task 3

- Prove that you are you,
- and the server is who you think it is!

Don’t reinvent the wheel
Open acknowledged standards FTW
Let someone review your processes

Summary

Baseline

Authentication

Authorization

Anomalies

Summary

Summary

Baseline	Divide (your design) and conquer!
	Ask SecOps for assistance in baseline security
	Code quality ahead of features
Authentication	
Authorization	
Anomalies	
Summary	

Summary

Baseline	Divide (your design) and conquer!
	Ask SecOps for assistance in baseline security
	Code quality ahead of features
Authentication	Don’t reinvent the wheel
	Open acknowledged standards FTW
	Let someone review your processes
Authorization	
Anomalies	
Summary	

Summary

Baseline	Divide (your design) and conquer!
	Ask SecOps for assistance in baseline security
	Code quality ahead of features
Authentication	Don’t reinvent the wheel
	Open acknowledged standards FTW
	Let someone review your processes
Authorization	Coming soon on Workshop 2 Part 2!
Anomalies	Coming soon on Workshop 2 Part 2!
Summary	Coming soon on Workshop 2 Part 2!

GÉANT
Networks • Services • People

Thank you

Any questions?

www.geant.org

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).