Signatures and receipts in the supply chain security

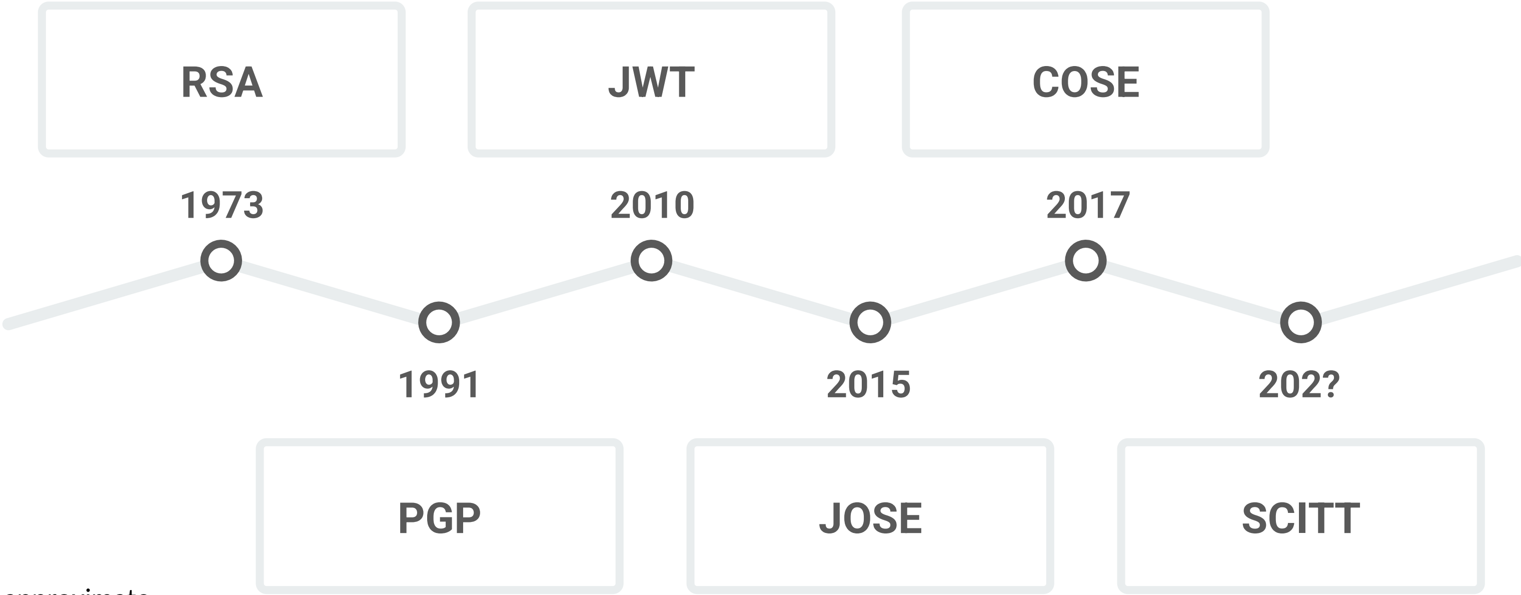
Thoughts are my own.
Do not quote me - Ivar

The public sector, gov agencies, and SMEs struggle with security after buying the software.

SCITT was born

- SUPPLY
- CHAIN
- INTEGRITY
- TRANSPARENCY
- TRUST

Evolution?



RSA 1973

PGP 1991

JWT 2010

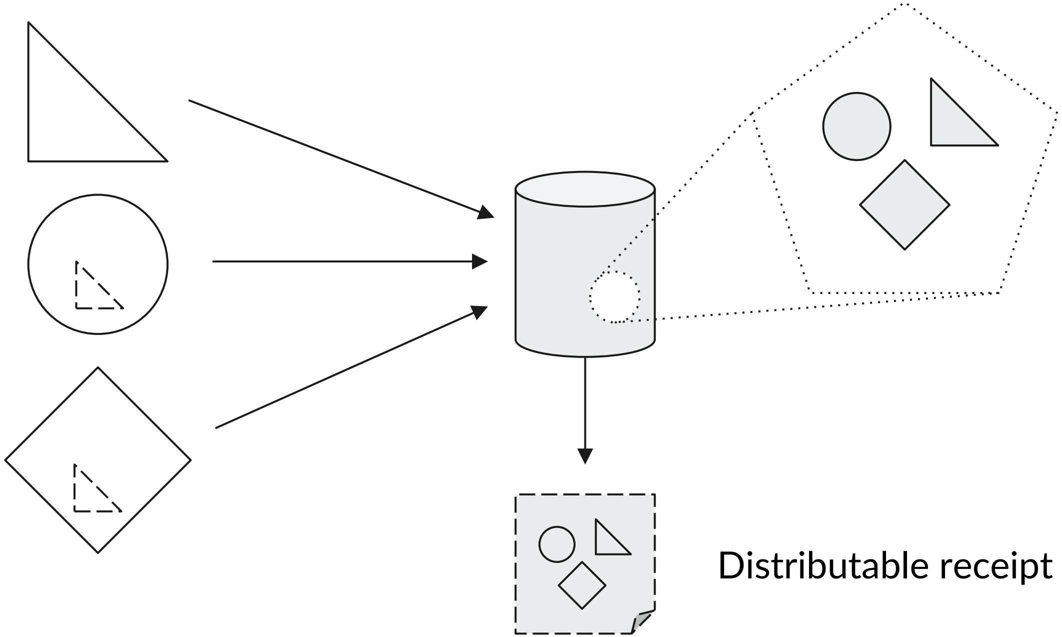
JOSE 2015

COSE 2017

SCITT 202?

* Dates are approximate

High level view

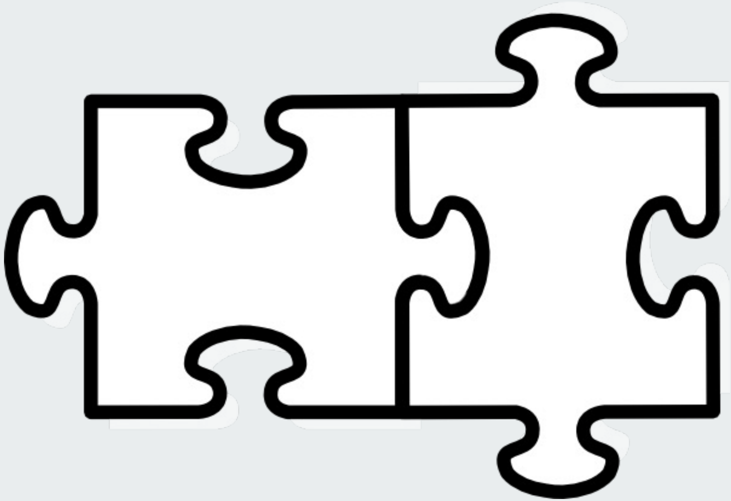


Signed payloads

Auditable storage

Distributable receipt

COSE



- An envelope similar to *JWT* (think *JOSE*)
- Smaller (including encoder, decoder)
- Can use bytes without *base64* or similar encoding
- Standardized countersignatures - *almost*

General structure

Signed payload (COSE_Sign1):

- Headers (protected, unprotected)
- Payload
- Signature
  - protected headers
  - payload

Countersignature:

- Headers (protected, unprotected)
- Signature:
  - protected headers
  - source protected headers
  - source payload
  - source signature

WAIT
WHAT IS THE PAYLOAD?
imgflip.com

Resist you must,
NIST* force is nearby

*National Institute of Standards and Technology

Challenges

- Different receipt formats
- Missing tooling
- Who runs it?
- How do you link receipts?

PAW
SHOP
LET'S FIX THIS!
imgflip.com

References

- Signature, receipt playground - https://playground-cose-eastus-api.azurewebsites.net/
- COSE RFCs: RFC9052, RFC9338
- About SCITT https://scitt.io
- About SBOM in NIST https://www.nist.gov/itl/executive-order-14028-improving-nations-cybersecurity/software-security-supply-chains-software-1