Using a certificate authority to manage signing keys

Magnus Bäck, Axis Communications

Eiffel

Why should we sign Eiffel events?

Structure of a signed event

```
{
  "meta": {
    "security": {
      "authorIdentity": "CN=joe,DC=example,DC=com",
      "integrityProtection": {
        "alg": "ES256",
        "signature": "b25zIDI0IGFwciAyMDI0IDIxOjQ0OjM0IENFU1..."
      }
    }
  },
  "data": { ... },
  "links": [ ... ]
}
```

Eiffel

Ecosystem support

.NET SDK (sort of)

Go SDK

eiffel-broadcaster Jenkins plugin

But which public key should we use to verify an event's signature?

Including the public key in the event payload

Bundling the public key in the event itself (meta.security.integrityProtection.publicKey) protects against corruption, but doesn't authenticate the sender.

If new publishers could send a "bootstrap" event with the key they intend to use this could be picked up and saved for later.

Introduce meta.security field for certificate

If the event includes a certificate signed by a CA accepted by the consumer, this will be enough to authenticate the event.

The certificate's subject would be required to match meta.security.authorIdentity.

Configuration in each consumer

Each consumer would be configured with a list of (event sender DN, public key) mappings.

Shared configuration

There's a common configuration source where all consumers can look up event sender DNs and get public keys.

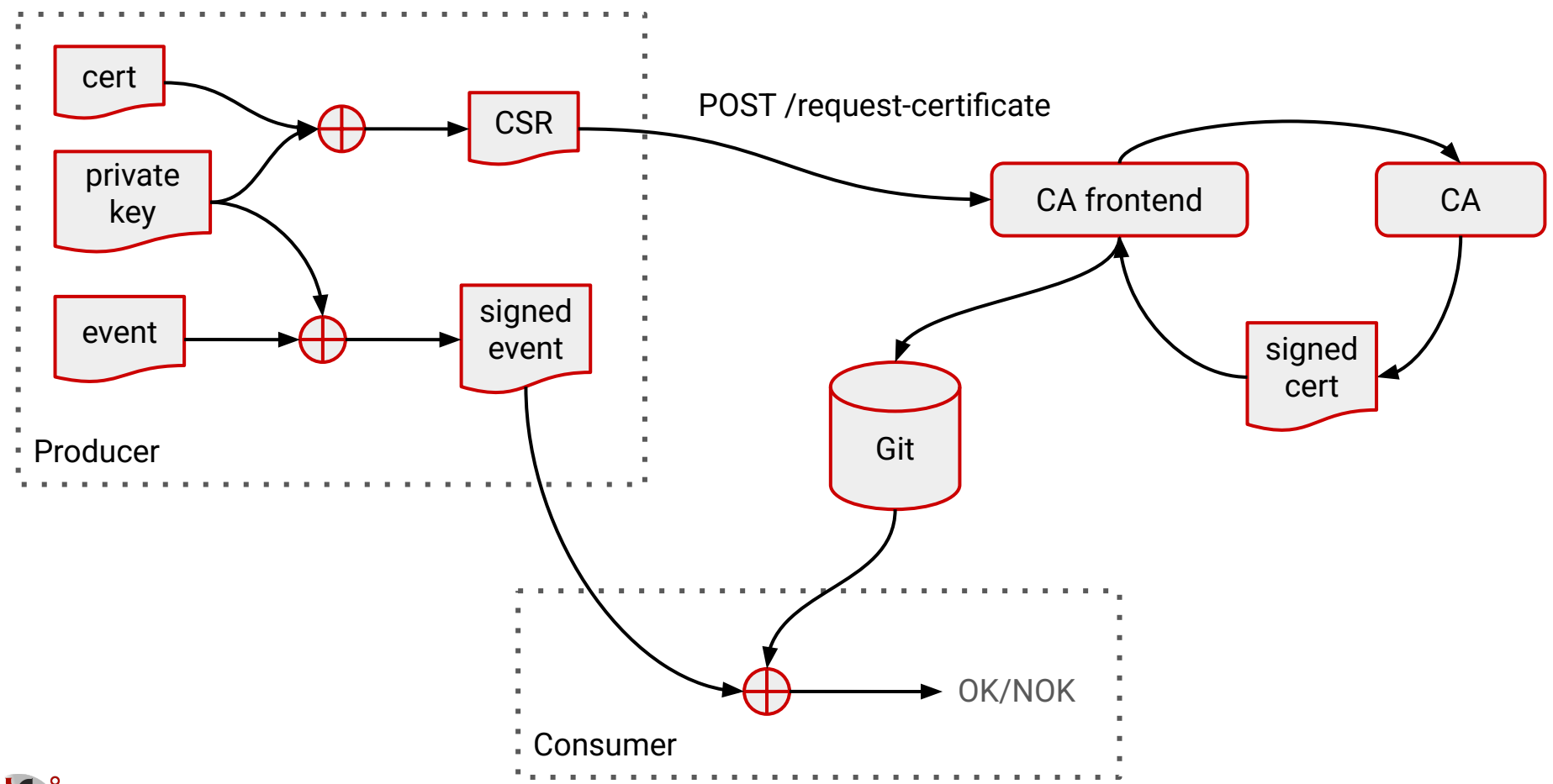
- One or more files in a git repository.
- A web service or similar API.

Drawback: S/he who controls the configuration source can impersonate anyone.

Shared configuration with CA-issued certificates

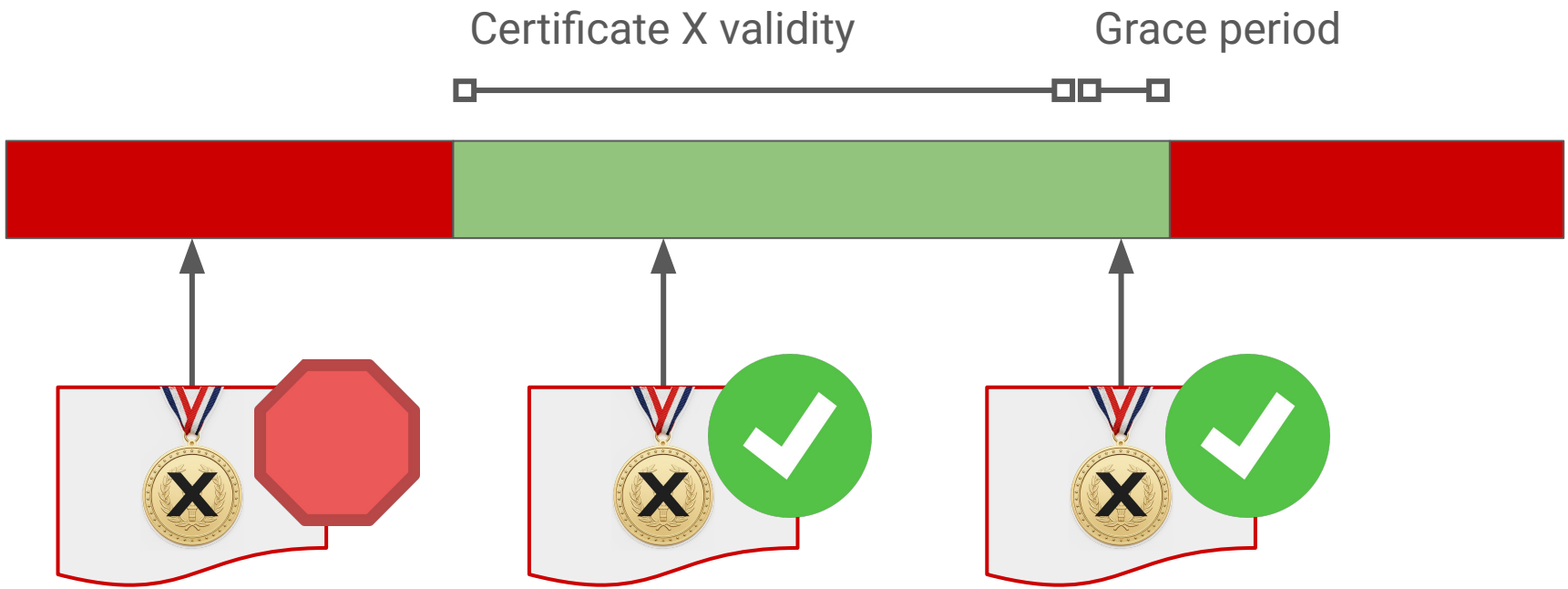
There’s still a common configuration source where all consumers can look up event sender DNs and get certificates with public keys.

Certificates are issued by a certificate authority that consumers trust.


cert
private key
event
CSR
signed event
Producer
POST /request-certificate
CA frontend
CA
signed cert
Git
OK/NOK
Consumer

Eiffel

Certificate validity considerations



Eiffel

Revoking a certificate

If a key is compromised, the CA's Certificate Revocation List (CRL) can be updated.

Its URL can be included in the issued certificate, and the data is signed by the CA.

Questions?