OpenDNSSEC Architecture

.se

What?

- OpenDNSSEC is a zone signer that automates the process of keeping track of DNSSEC keys and the signing of zones.

.se

Why?

- The available DNSSEC tools were lacking:
  - Good key management
  - Policy handling
  - Hardware acceleration
  - Etc.
- DNSSEC should be easy to deploy
- Increase the number of DNSSEC users
- Experience from previous DNSSEC operations

.se

Who?

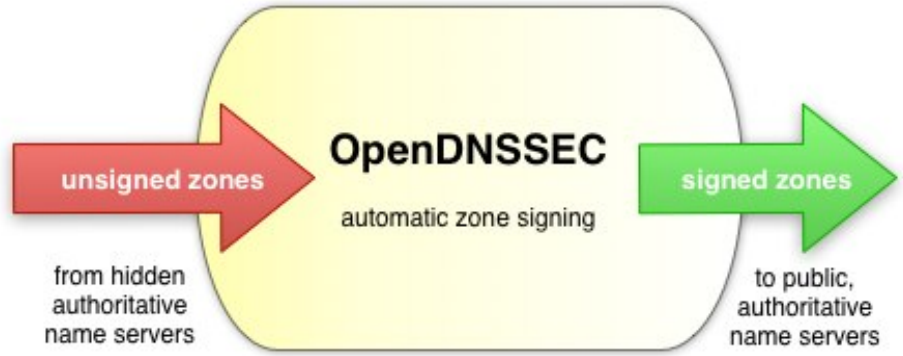
.se

About OpenDNSSEC

- Simplifies the process of signing one or more zones
- Reducing the work load on the system administrator
- Open source software with a BSD license
- Simple to integrate into existing infrastructure
- Key storage and hardware acceleration using PKCS#11

.se

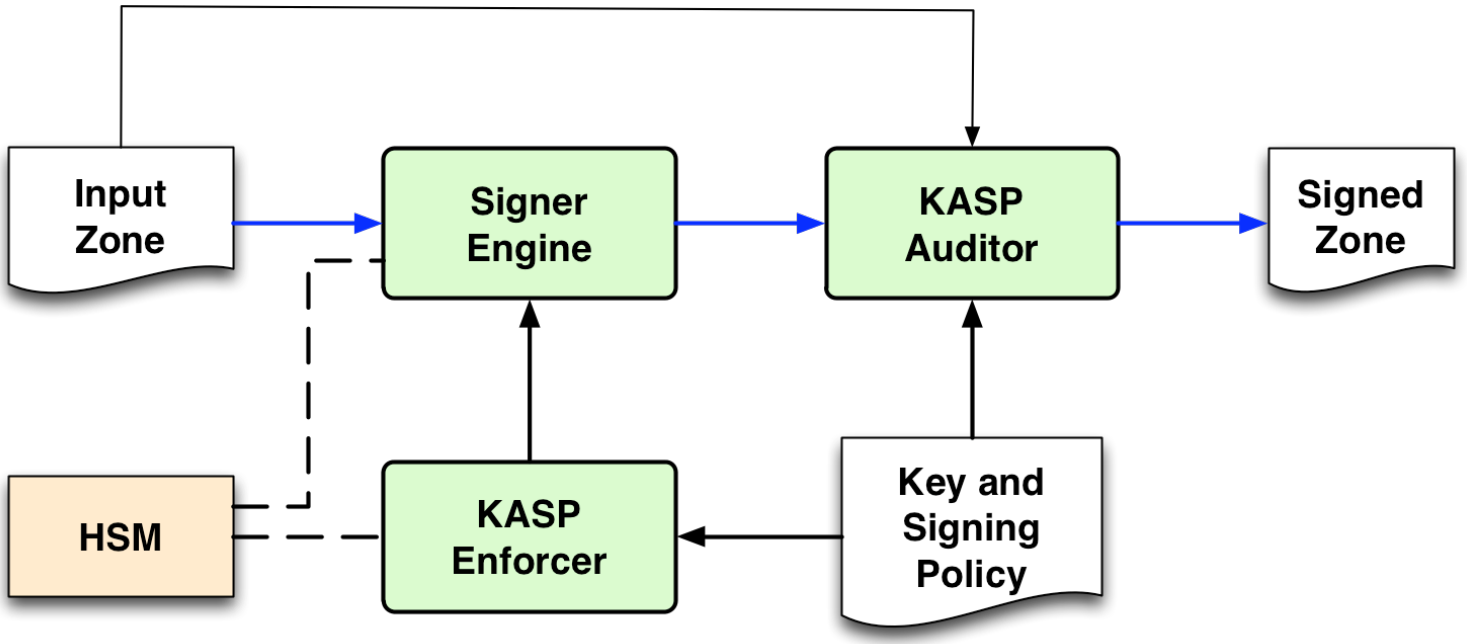
Bump-in-the-Wire



- In many cases, anticipate that OpenDNSSEC will be employed on a system between a hidden and public master.

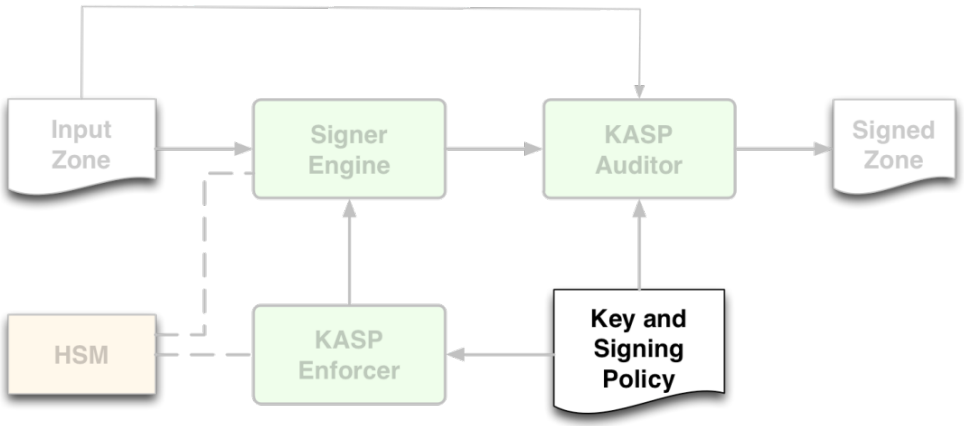
.se

Architecture



.se

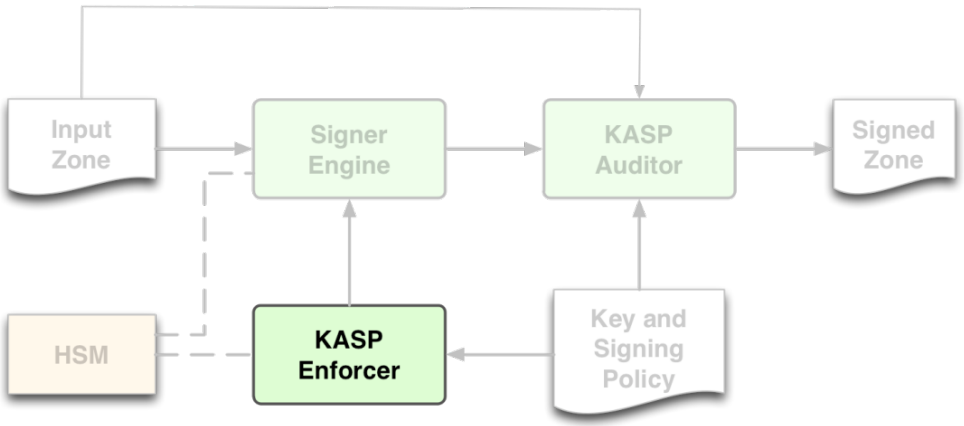
Key and Signing Policy



- How to sign a zone is described by a policy
- Allows choice of key strengths, algorithm, key and signature lifetimes, NSEC/NSEC3, etc.
- Can have anything between one policy for all zones to one policy per zone.

.se

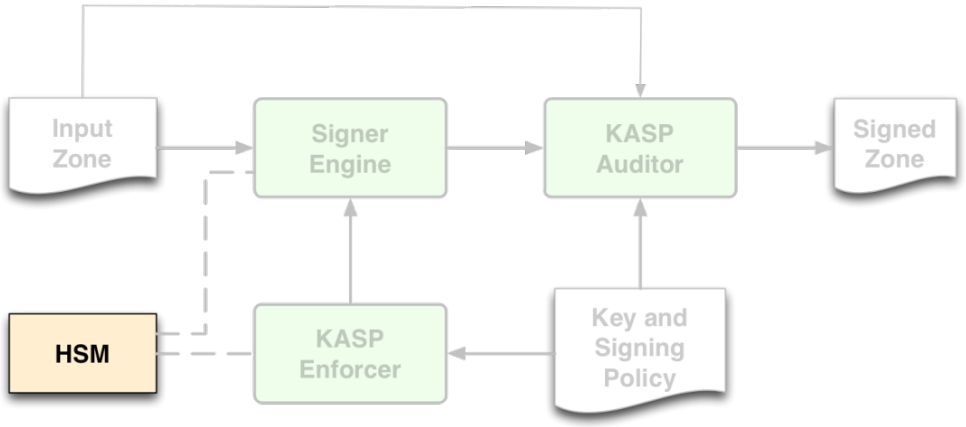
KASP Enforcer



- Handles the management of keys:
  - Key creation using HSM
  - Key rolling
- Chooses the keys used to sign the zone.

.se

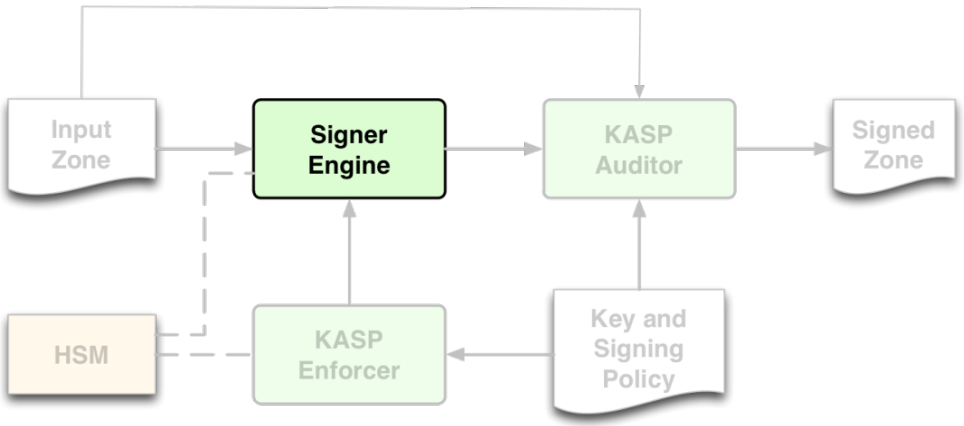
HSM



- Hardware Security Module
  - Stores the keys
  - Hardware acceleration to sign records
- Standard interface via PKCS#11 API
- SoftHSM available with OpenDNSSEC

.se

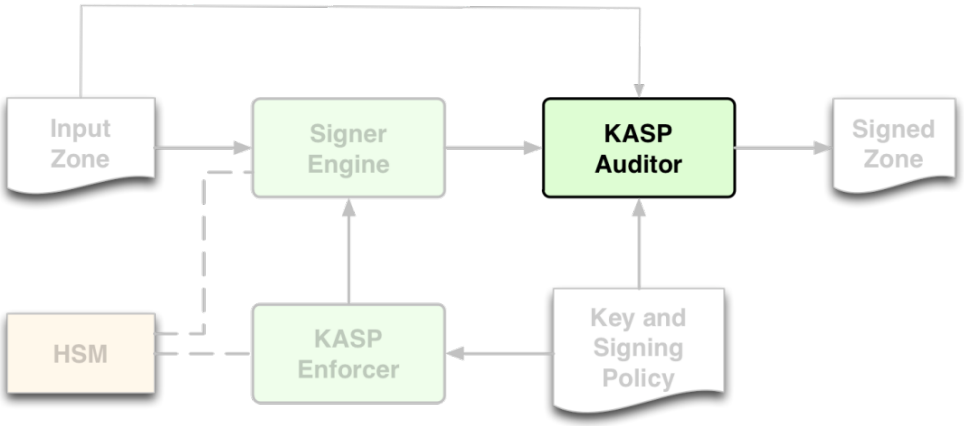
Signer Engine



- Automatic signing of the zones
  - Can reuse signatures that are not too old
  - Can spread signature expiration time over time (jitter)
- Maintains the NSEC/NSEC3 chain
- Updates SOA serial number

.se

KASP Auditor



- Checks that the signer and enforcer work the way they are supposed to, e.g.
  - Non DNSSEC RRs are not added or removed
  - Policy is being followed
- Can stop the zone distribution if needed
- Written in Ruby

.se

More about HSMs

Why should you use one?

- Security (FIPS)
  - The private keys are stored securely in the HSM
  - You know where your keys are
- Speed
  - 1 – 13,000 signatures per second

Are they expensive?

- $50 - $25,000

Remember to protect the host

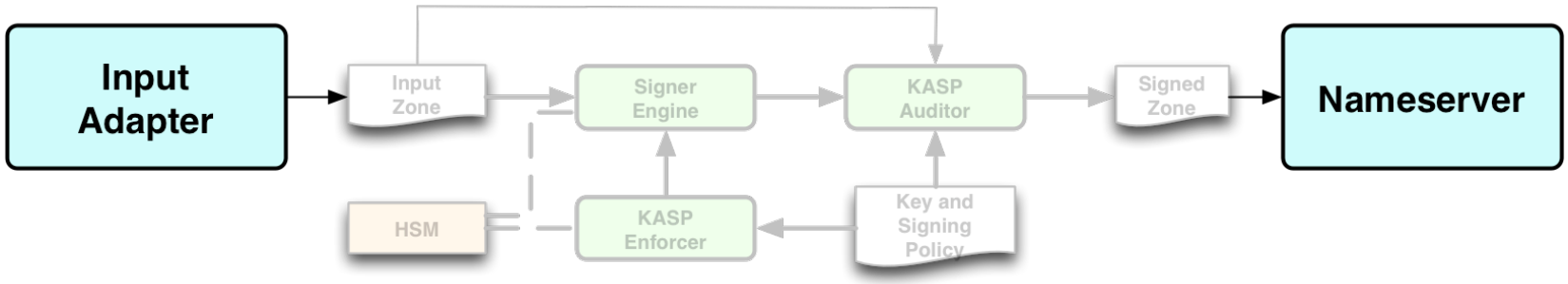
- Garbage in -> Garbage out

SoftHSM

- SoftHSM is a software-only implementation of an HSM using the PKCS#11 interface
- Can be used to test the PKCS#11 interface without buying a real HSM.
- SoftHSM is developed as a part of the OpenDNSSEC project.
- Uses Botan and SQLite.
- SoftHSM makes it possible to use OpenDNSEC in a software-only environment.

.se

Input and Output Adapters



- Input adapter supplied as part of OpenDNSSEC - accepts AXFRs, responds to NOTIFYs.
- Output adapter not supplied - any preferred nameserver can be used (BIND, NSD, etc.)
- Can configure command to be used to reload zone.

.se

Configuration

- You need an HSM. (E.g. SoftHSM)
- Uses XML-files for configuration
  - conf.xml
  - kasp.xml
  - zonelist.xml
  - zonefetch.xml
  - eppclientd.conf

.se

Status

- 1.0 alpha released in July
- 1.0 beta released in October
- 1.0 released in February

Release plan

- 1.1 Increased speed (April)
- 1.2 New internal engine (June)
- 1.3 Better flow of information (August)

.se

Development

- Modular nature
- Each participant could work separately
- Agree the interfaces
- Regular phone conferences
- Code checking with Coverity Prevent
- Code reviews before releases

Testing

- Modular nature
- Each module has unit tests
- Functional testing performed by SIDN
- Performance testing by SIDN and .SE

Thank you

- http://www.opendnssec.org/

SoftHSM

.se