draft-johani-tld-zone-pipeline-00

Requirements and Design Consequences For a TLD Zone Pipeline

Johan Stenstam Jakob Schlyter

July 22, 2023

Problem Statement

The .SE Registry had a nasty incident in February 2022. We published the .SE zone with several thousand DNSSEC signatures that did not validate.

- This is still strange, because we had two independent HSMs that apparently failed at the same time and the vendor has been unable to explain this. However, that is not the topic of this presentation.
- Regardless of failing HSMs, we **could** have detected that the zone contained invalid data before publication. But we didn't.
- The reason was simple: "**Don't touch it, it works.**"

In the aftermath of this incident we decided to reimplement the entire zone pipeline from scratch. But before that we should go back to first principles:

- **Define the requirements.**

The Purpose of this Document

Zone production for a TLD (or other seriously important zone) is not rocket science. But it is. . . **important to get absolutely right**.

- But with millions of delegations, DNSSEC signing, etc, how should "right" be defined?
- Not to mention verified before publication.

Perhaps part of getting the zone right is achieved by ensuring that the zone production is designed in a way that makes it as difficult as possible to get it wrong.

We decided to focus on basic principles for zone production rather than exactly what verification tests should be performed.

- **Tests** will change over time, tests will differ between zones.
- **Basic principles** will hopefully be more generally useful.

Why Make This An Internet-Draft?

The reason is that the discussion about what the requirements on the zone generation should be is not particular to the Swedish Registry. It is a generic discussion that all TLDs (and other important zones for that matter) should have.

- The intent with the document is to encourage an open discussion among TLDs on these issues.
- Secondly, the new, modular, implementation of the zone pipeline for `.SE` is currently in testing and will become open source.

The end goal is to try to use the serious **wake-up call** that we had a year ago and try to turn that into a combination of **requirements**, **design consequences** and **actual code** that may help also other systemically important zones with their zone generation pipelines.

Let's Get Back To: Basic Principles

The zone pipeline consists of several steps. These are our principles for the zone pipeline:

- Data transport between steps should use open, standardised protocols (eg. DNS AXFR/IXFR).
- The critical path for zone data follows must be via well-reviewed standard software (i.e. no custom software).
- Bugs in custom software must not be able to affect zone content.
- Errors in the input must not be able to negatively affect the ongoing publication of (an earlier version of) the zone.

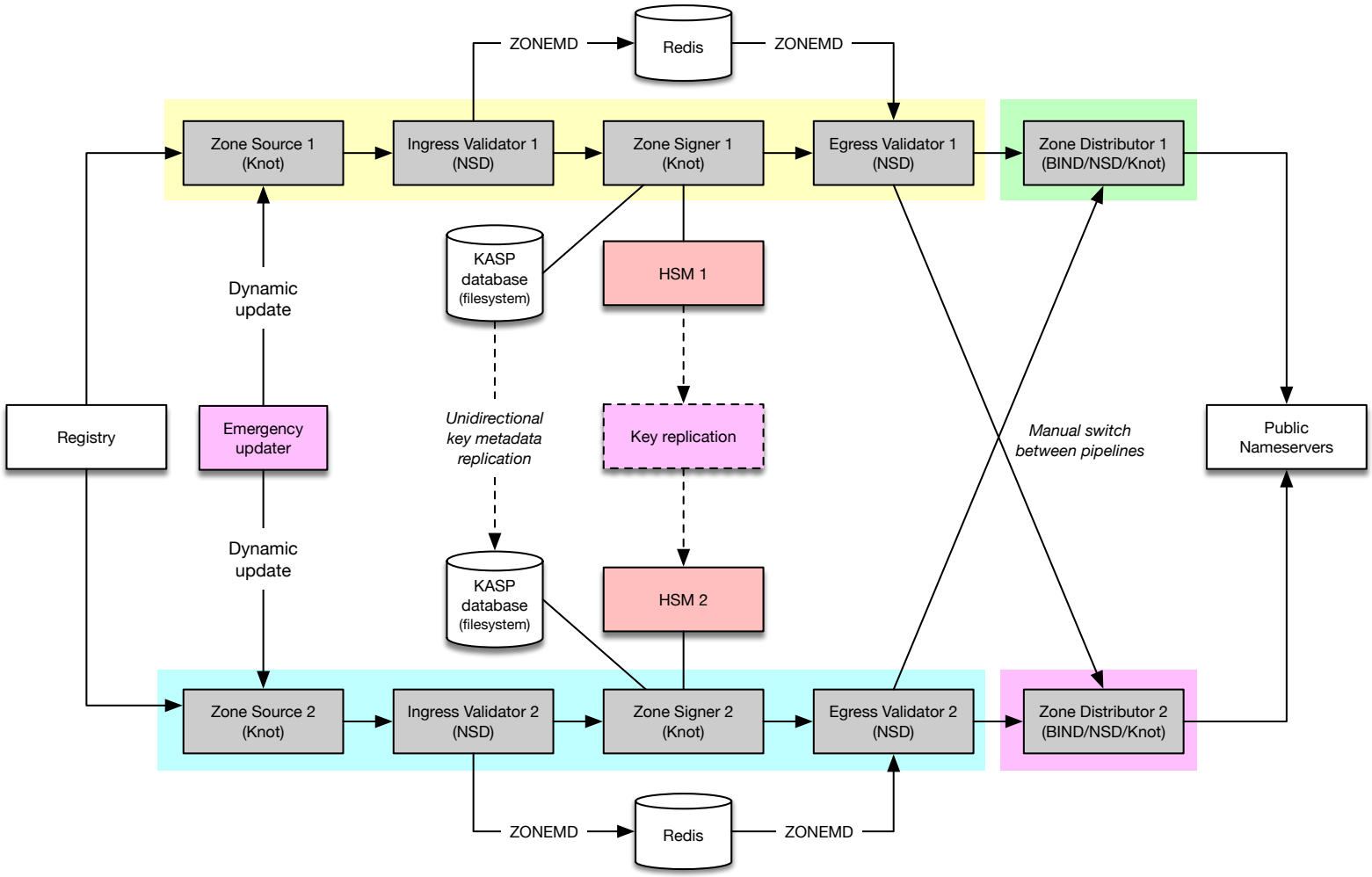
Let's Get Back To: Basic Principles

"And if you don't like them, I have others"
—Groucho Marx

The zone pipeline consists of several steps. These are our principles for the zone pipeline:

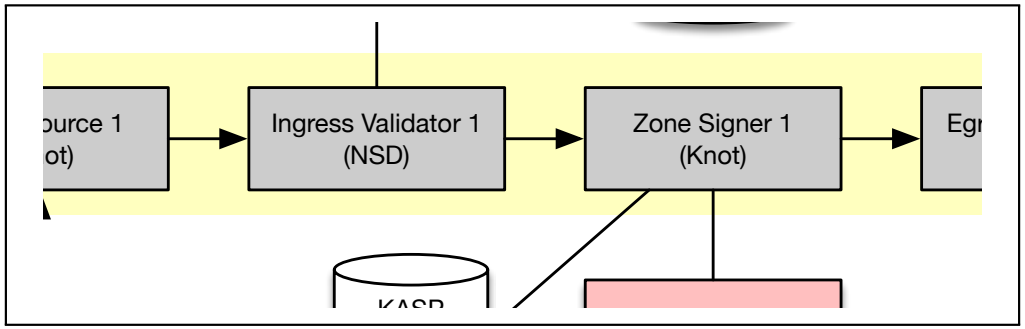
- Data transport between steps should use open, standardised protocols (eg. DNS AXFR/IXFR).
- The critical path for zone data follows must be via well-reviewed standard software (i.e. no custom software).
- Bugs in custom software must not be able to affect zone content.
- Errors in the input must not be able to negatively affect the ongoing publication of (an earlier version of) the zone.

Resulting Design Overview



Design Consequences: Validation of the Unsigned Zone

Validation must be done on the unsigned zone, because if a showstopper error is discovered it is too late to roll back after the flawed zone has been signed.



Examples of policy requirements:

- Critical parts of the zone (the apex in particular) must match exact criteria.
- Changes to dynamic parts of the zone (i.e. 99.9% of the records) must not exceed a defined threshold between two versions of the zone.

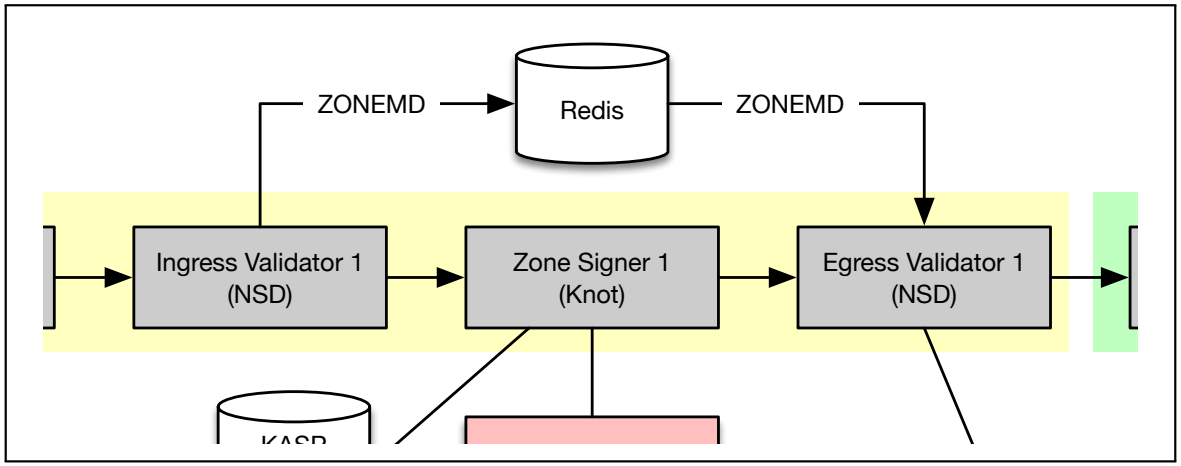
Validation of the Signed Zone

It is one thing to define requirements on the unsigned zone from the registry.

It is a different thing to define requirements on the signed zone. Obviously all DNSSEC signatures must validate.

- But then? How to verify that the "signing operation" has not made any other changes to the zone other than adding millions of DNSSEC records?

Design Consequences: Validation of the Signed Zone



By saving a checksum of the unsigned zone (`ZONEMD` is designed for this) it is possible to define two requirements that the signed zone must fulfill:

- Every added `RRSIG` (and `NSEC`, etc) must verify correctly.
- If all the DNSSEC data is removed from the signed zone then the result should have exactly the same checksum as the unsigned zone, i.e. the `ZONEMD` must match.

More Design Consequences

1. Because "**tests**" (which are typically custom software) must not be in the critical path. Therefore we need the ability to do a callout to an external `verifier` from standard software (a nameserver) in the critical path.

   At the moment we use the `verify:` attribute in NSD for this, but we hope there will similar support in other software in the future.
2. I.e. when a new zone is received in one of the validator steps it will not propagate further, unless the callout to the external verifier returns a positive result.

Implementation Status

- Tests have a standard interface to make them trivial to add to (or remove from) the `verifier` framework.
- Tests are either **internal** to the `verifier` or **external** (eg. tests that run an external checker like `dnssec-verify`).
- The "`verifier`" is a static Go binary (we prefer compile time dependencies over runtime dependencies).
- The implementation is functionally complete and is currently in testing in parallel with the existing old system
- The "interesting" parts will be made open source for other interested parties to use, pick parts from and, hopefully, contribute to.
  - Things like the exact interface to our registry system, Kubernetes provisioning configs and the emergency updater, will be excluded.

Anyone interested should talk to me.

Document Status

The document is also functionally complete, but it presently has, at least, two flaws:

- While we've tried to make it as generic as possible it seems likely that we've failed here and there. It would be great with input from others to make this a more collaborative effort resulting in wider applicability.
- It is in need of much more third party review.

Therefore we would like to see this document adopted by the DNSOP WG.

Contact Information

Johan Stenstam	johan.stenstam@internetstiftelsen.se
Jakob Schlyter	jakob@kirei.se