Speedup – High performance Signing-Cluster for big zones

Christian Petrasch, DENIC eG

DNS-OARC – Austin – 2019-10-31

Motivation

- Speedup zone updates

  (query a new created or updated domain in a few minutes)

- Multi-client capability of the system for other domains then .de

  (customers or denic.de)

- Simplification or automation of

  key management

- More cost-effective keystore

  (HSM) solution

Imagesource: ideenexpo.de

Status Quo..

- One zone update per hour
- Sign the whole zone every time

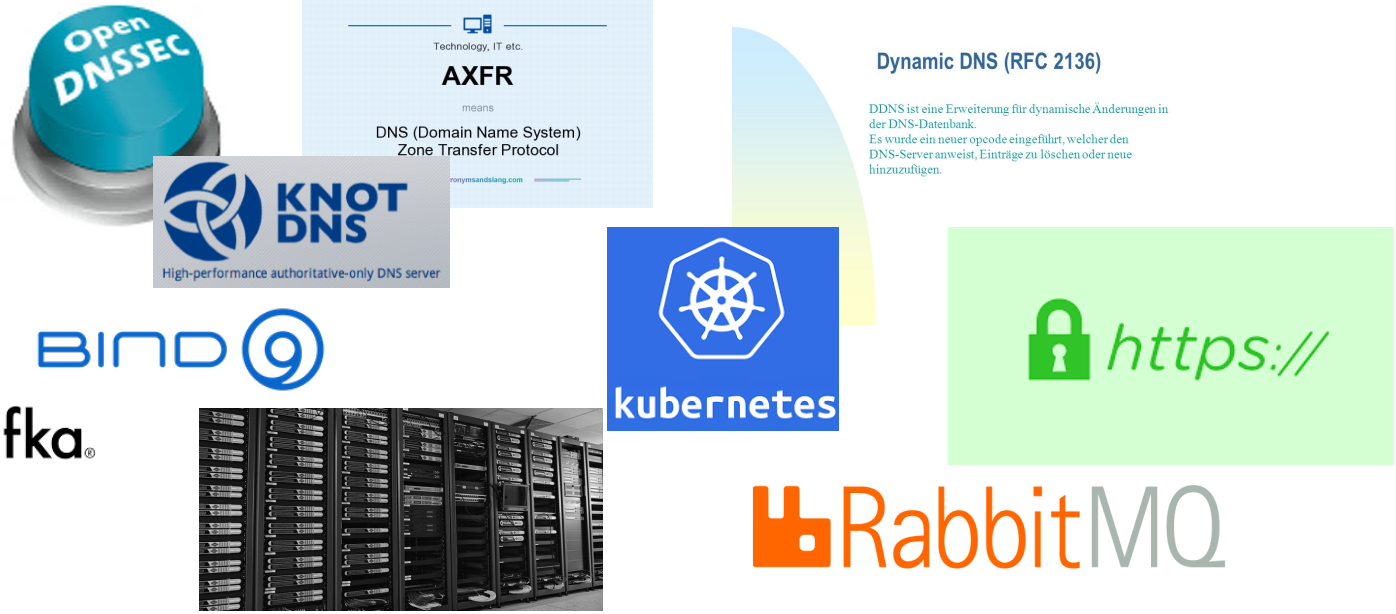
  (big increments to transfer)

- Proprietary signing solution
- Key management based on manual key ceremony analogous IANA offline key ceremony

Imagesource: de.dreamstime.com

Wishes and decisions..

- Sign incrementally
- Communication via standard protocols of DNS industry
- Usage of a community based signing solution (open source)
- Cluster solution (redundancy)



Decisions made...!

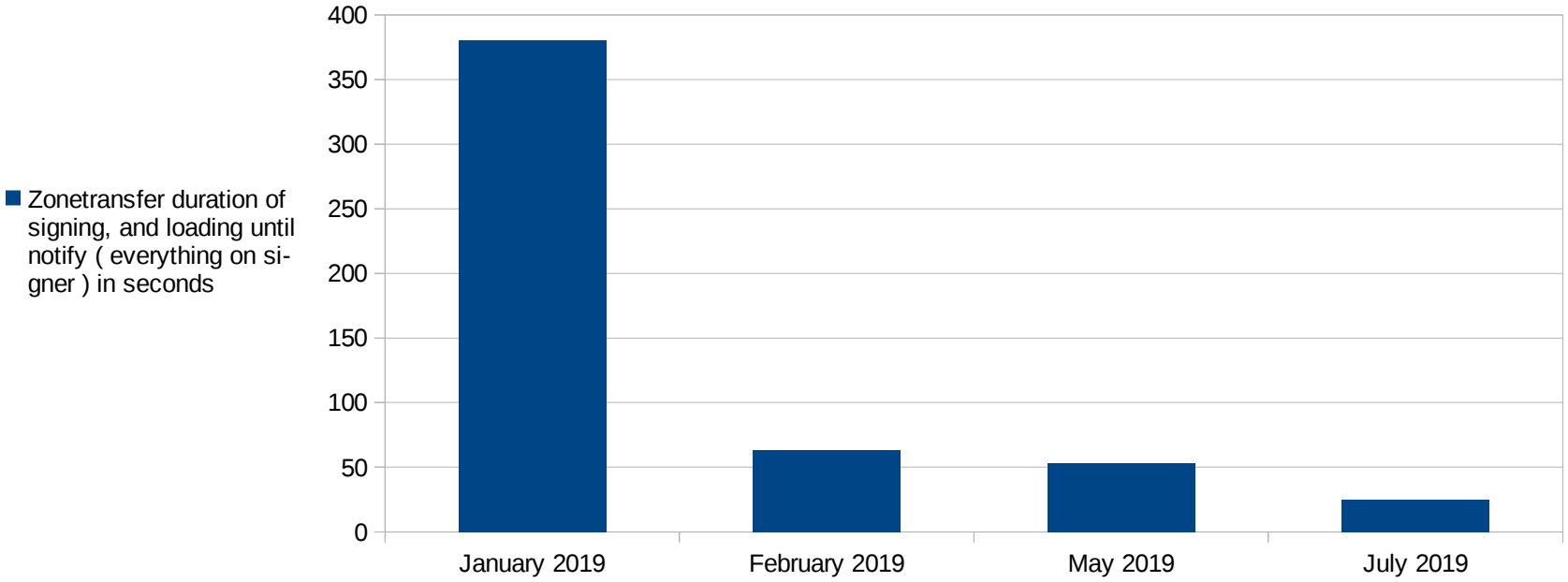
- For security reasons: On Premise
- Cluster: Kubernetes
  - Container Technologie has a lot advantages for developing Microservices
  - Good redundancy and failover features
  - Cloud migration possible

- Communication: Python Tool (JSON Messages) und DDNS
  - Python Tool for generating JSON Messages out of database
  - DDNS protocol provides API to change signing solution

- Signingsolution: KNOTdns
  - Good vendor and community supported

Why KNOT .. ?

- Signing Solution: KNOTdns
  - Really good support by CZnic
    - Tremendous performance boost for zone transfer since February 2019 (at version 2.9 / boost was required by DENIC)
  - All measurements are based on our .de zone with around 40 million entries



Why KNOT .. ?

- Signing Solution: KNOTdns

  - Bugfixing in direct communication just in time
  - In use at RIPE NCC, good experiences shared
  - Good Usability and easy to configure

- At this point:

  Special big thanks and greets to

  Daniel Salzmann and Libor Peltan from CZnic

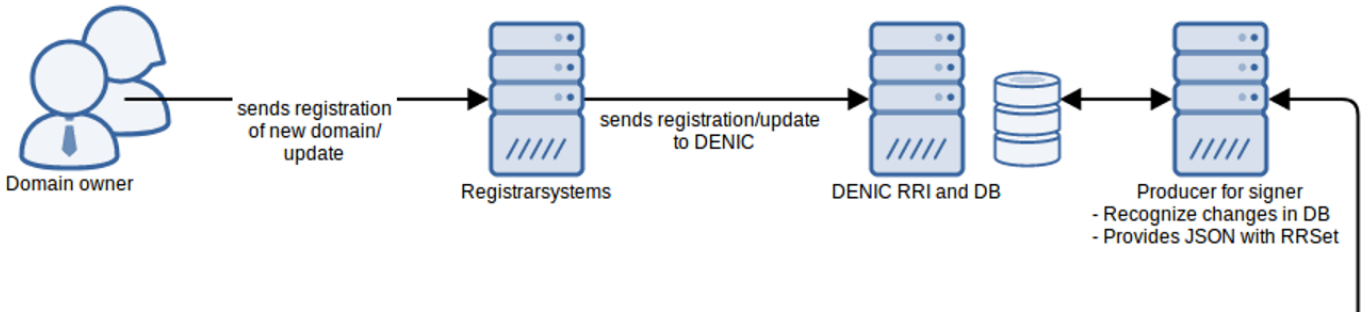
A lot of challenges … made !

- Queueing solution developed, MQueue doesn‘t fit perfect
  - The MessageQ, especially Kafka in Kubernetes wasn‘t easy to implement.
  - Good solution MQ but overkill for our setup so business value was negative in our purpose
  - Solved with changing the direction of the data stream (Poll instead Push mechanism) / Queueing in database
- Synchronisation solution for key material
  - GlusterFS based solution

From registry system..

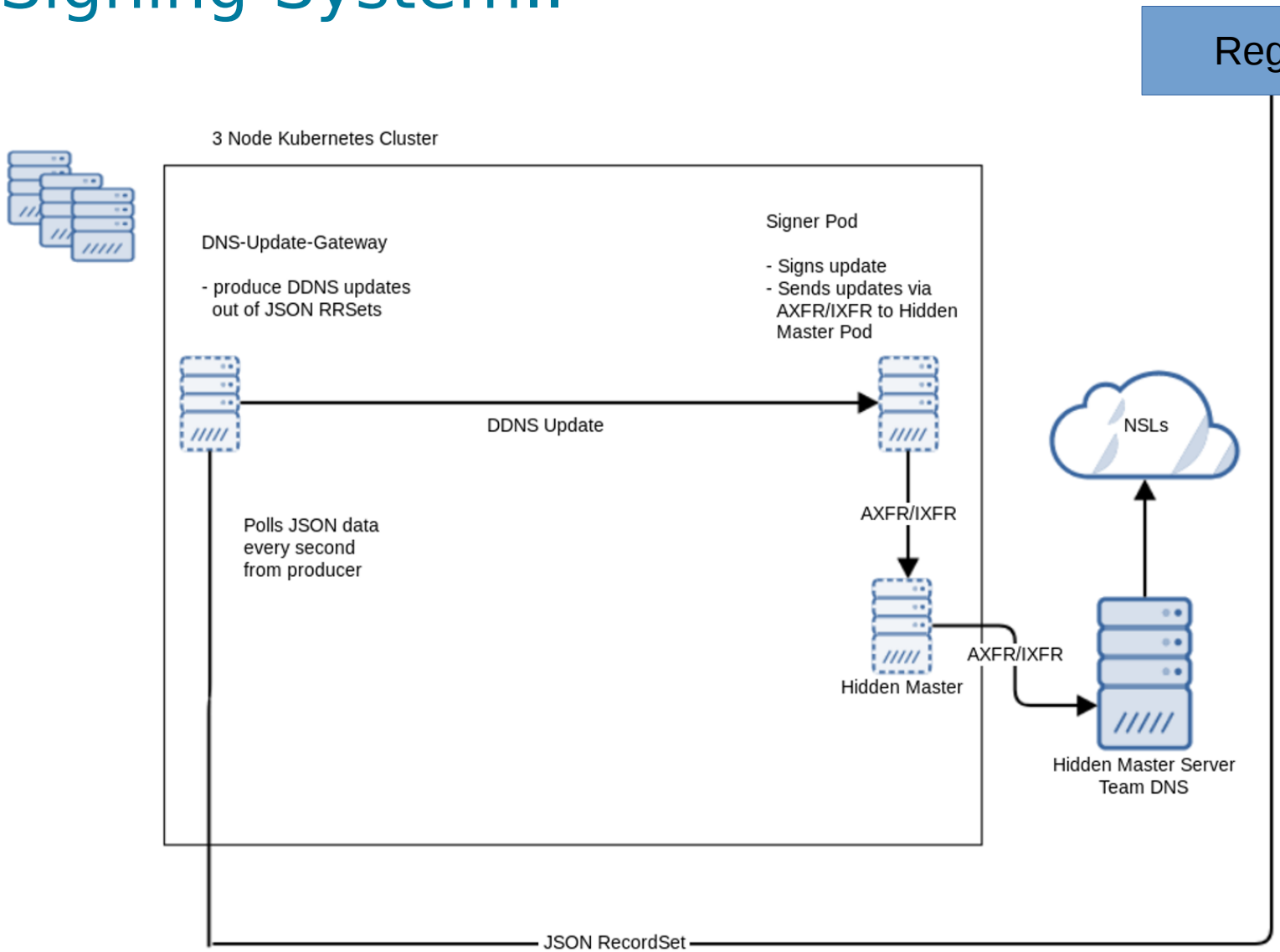
- Python Tool provides database-updates for signing system

- One update consists of a complete Record Sets of all data for the updated or created domains in JSON Format



Signing-Cluster

to Signing-System..



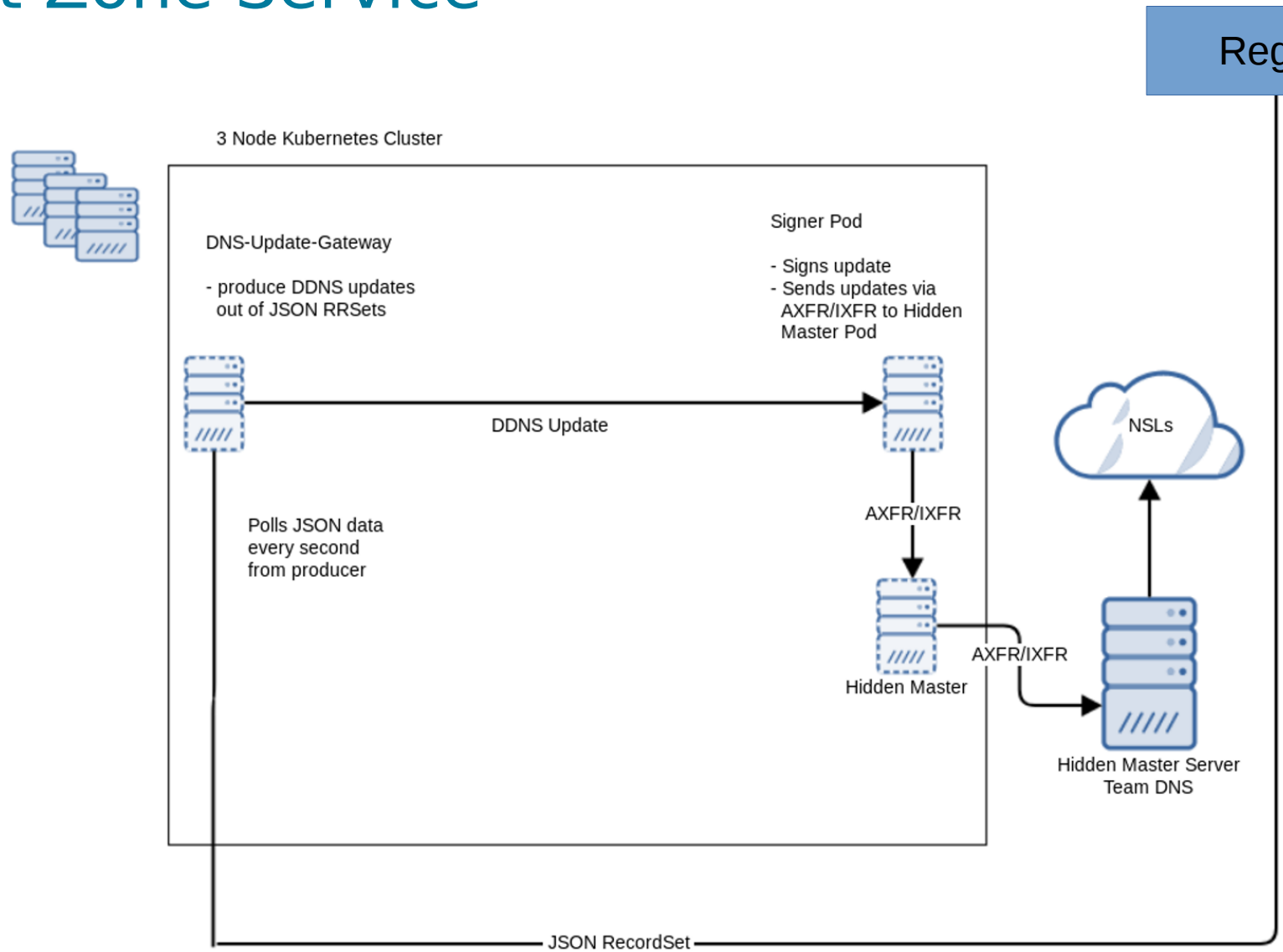
To Signing System .. 2. challenges made!

- Init Zone Service
  - Challenge:

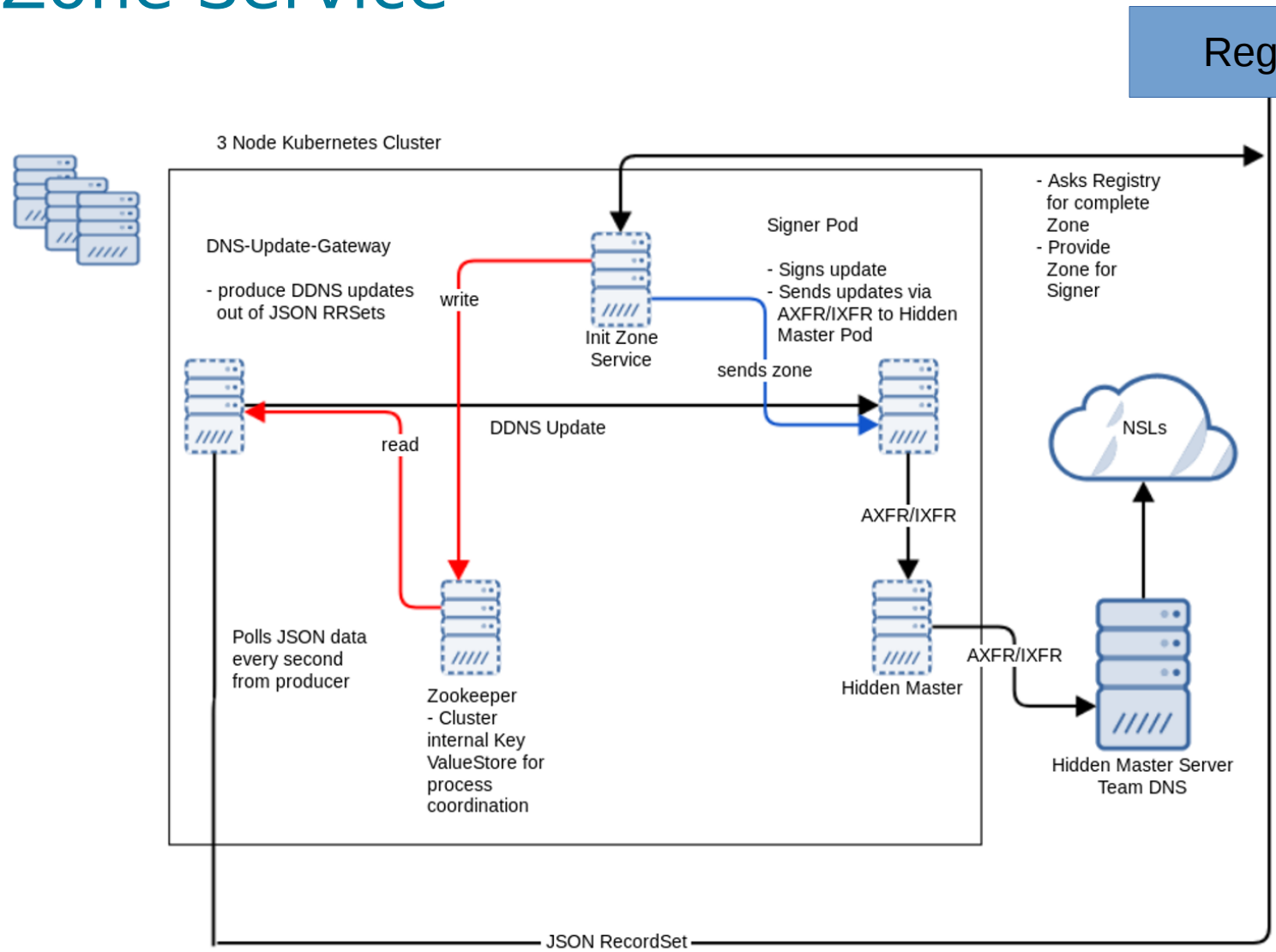
    Develop a tool which can provide the whole zone data in one update

  - Why?
    - The zone update mechanism for incremental updates is limited by around 1000 updates per JSON Message
    - Initial filling would take to long

Init Zone Service



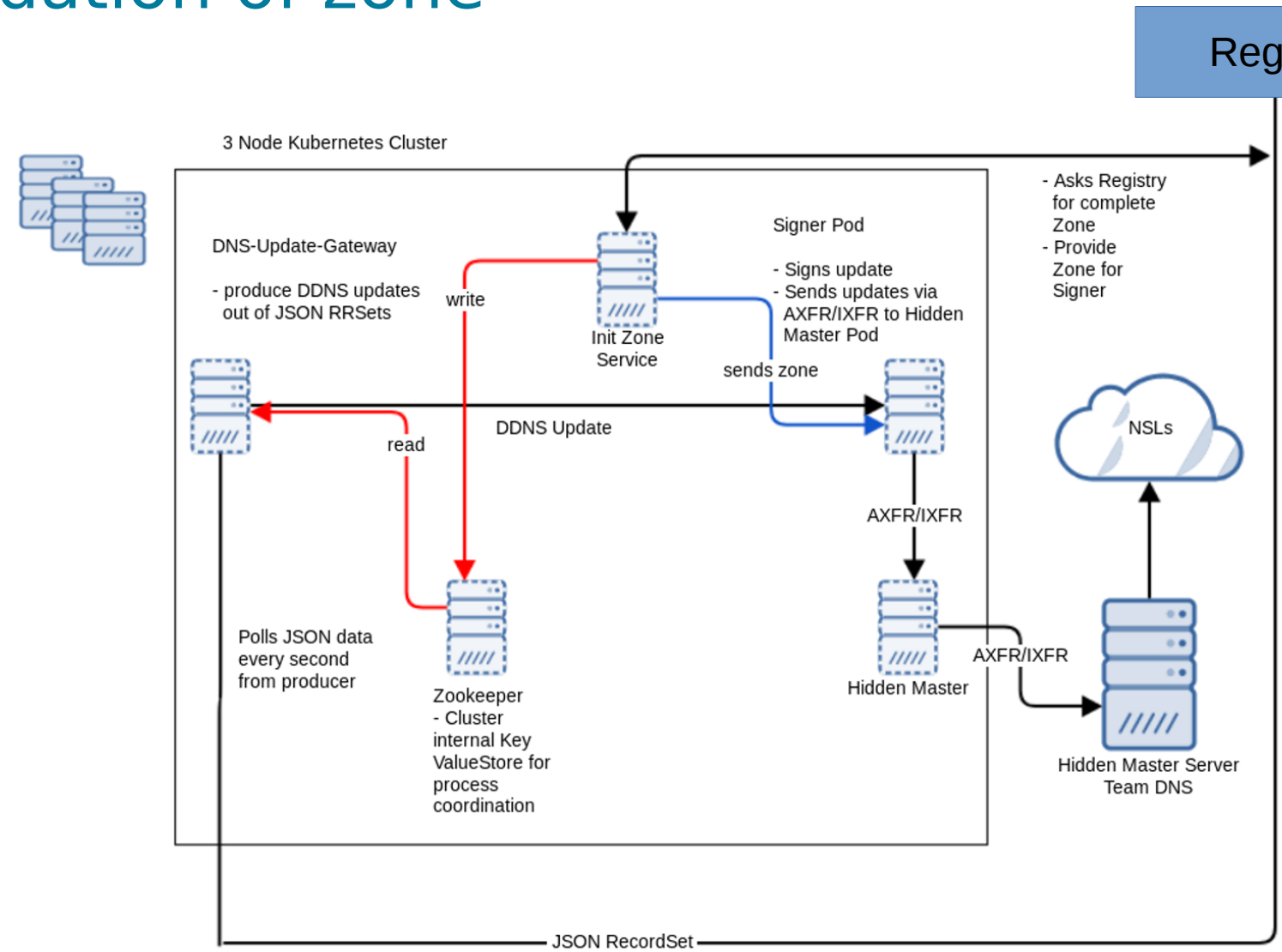
Init Zone Service



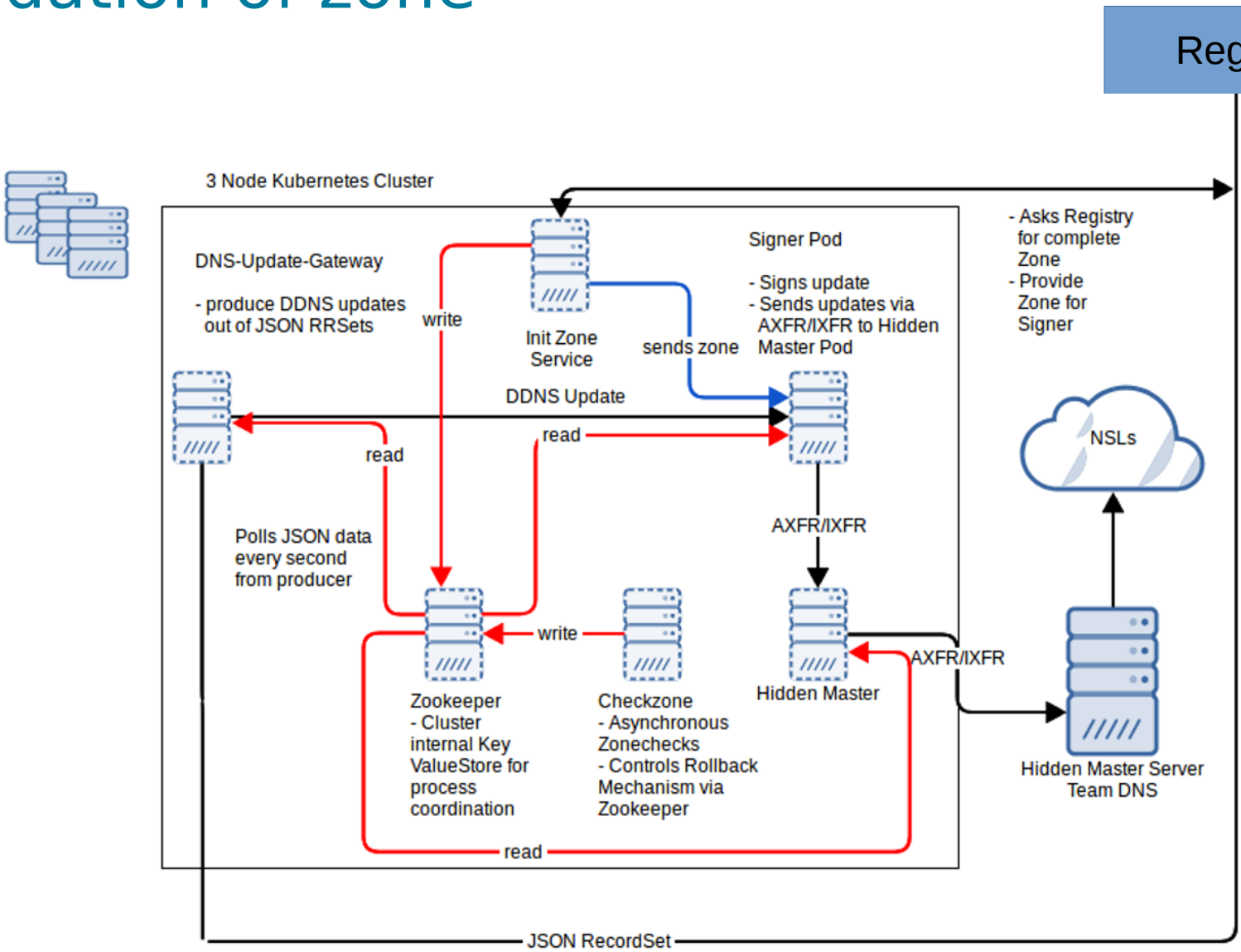
To Signing System .. 3. challenges made!

- Asynchronous zonevalidation
  - Asynchronous zonevalidation every 30 Seconds
  - Validationerror:
    - Stop zone deployment and trigger automatic reset of Zone
    - Check new zone and re-activate zone deployment if zone validation successful
    - Roll-forward !!
    - Parallel send alarm to emergency service to analyse issue
  - There is no synchron solution for validate incrementally at the market at this time. All validation tools work file based. With that and our big zone we can‘t reach our timing parameters. (But there is sth. in work)

Validation of zone



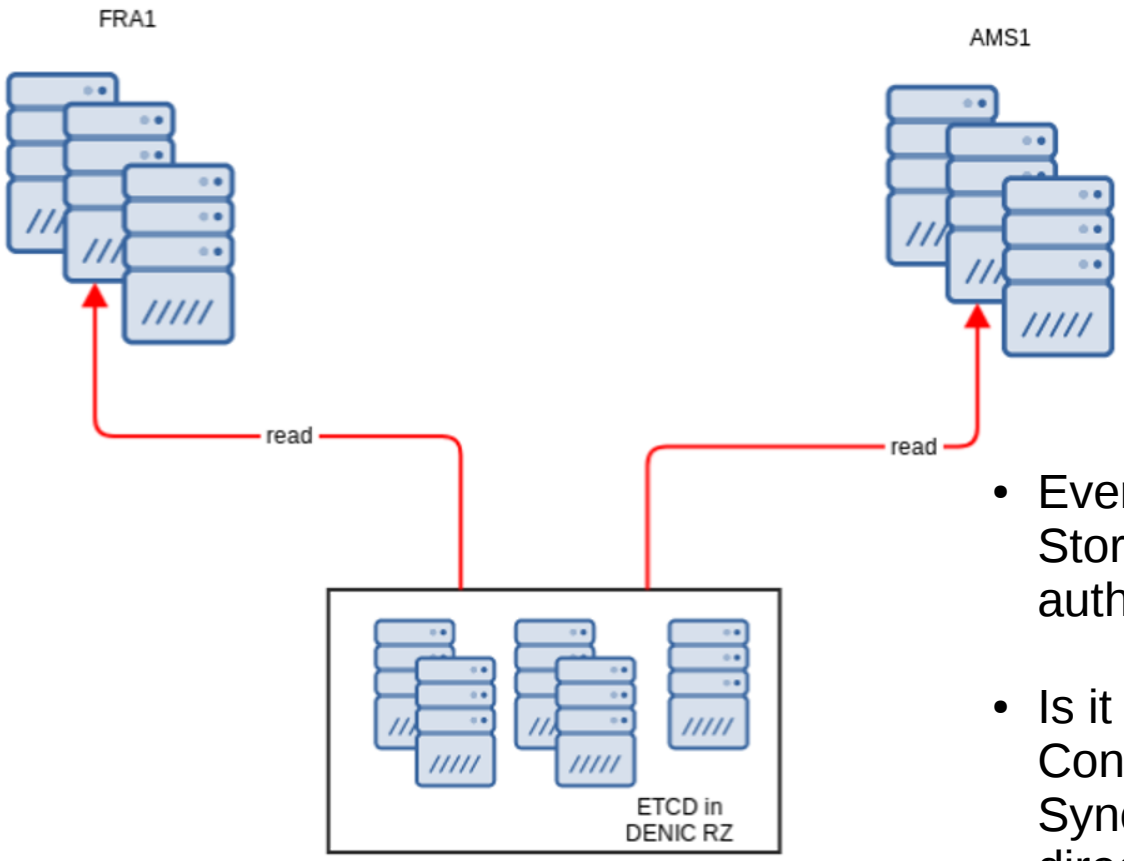
Validation of zone



To Signing System .. 4. challenges made!

- Controlmechanism for different situations of service
  - Challenge: Kubernetes Cluster are not optimal for stateful applications and active-passive services
  - Need: Switch between datacenters
  - Key-Synchronisation (Only on signer is allowed to be authoritative for key material)
- Solution: Key/Value Store, which provides data to the control mechanism of the cluster which cluster is authoritative (ETCD)

Redundanz and Failover



- Every cluster reads Key/Value Store to decide if it‘s the authoritative one
- Is it activ it starts the Signing-Container and the Key-Synchronisation in the direction to the passive cluster

The HSM Solution

- Requirements :
  - High performance
  - Sufficient secure

    (The security assessment was made under the criteria of the legacy signing platform and performed by a penetration test company)
  - PKCS11 capable
  - If possible, economical

The HSM Solution

- Survey of RIPE in community has revealed :
  - A lot of people think that HSMs are overkill (verbal information RIPE NCC)
  - RIPE itself no longer use HSMs, but they have made its signing systems more secure by hardening, firewalls and role concepts. RIPE do not use HSMs despite the use of a private key signing key in permanent access (online KSK).
- Survey of DENIC in the DNS Community revealed:
  - Surveyed registries mostly have „Offline KSK“ without HSMs or „Online KSK“ with HSM.

The HSM Solution

- After market analysis of the products :
  - There are only two products that are on interest for DENIC
    - SoftHSMv2 (SoftwareHSM with AES encryption)
    - SmartCardHSM (USB Token)

- After evaluating the security criteria, the decision was made for the SoftHSMv2.
  - Was rated as sufficiently safe with appropriate further measures
  - Is fast enough (this is not certain for the USB tokens)

Imagesource:
https://www.opendnssec.org/softhsm/

Security

- Role concept, similar to a DNSSEC key ceremony
- Firewalls / Separated networks
- Monitoring

- And further measures to secure the keymaterials, which can not be discussed for safety-related reasons

Imagesource: internetofbusiness.com

Demo

Thanks

Questions
?

contact:

petrasch@denic.de