The *Erik Synchronization Protocol* for use with the RPKI

Job Snijders
job@sobornost.net

I am Job Snijders

Hacker

Hiker

IETF RFC author (BGP & RPKI)

OpenBSD Developer

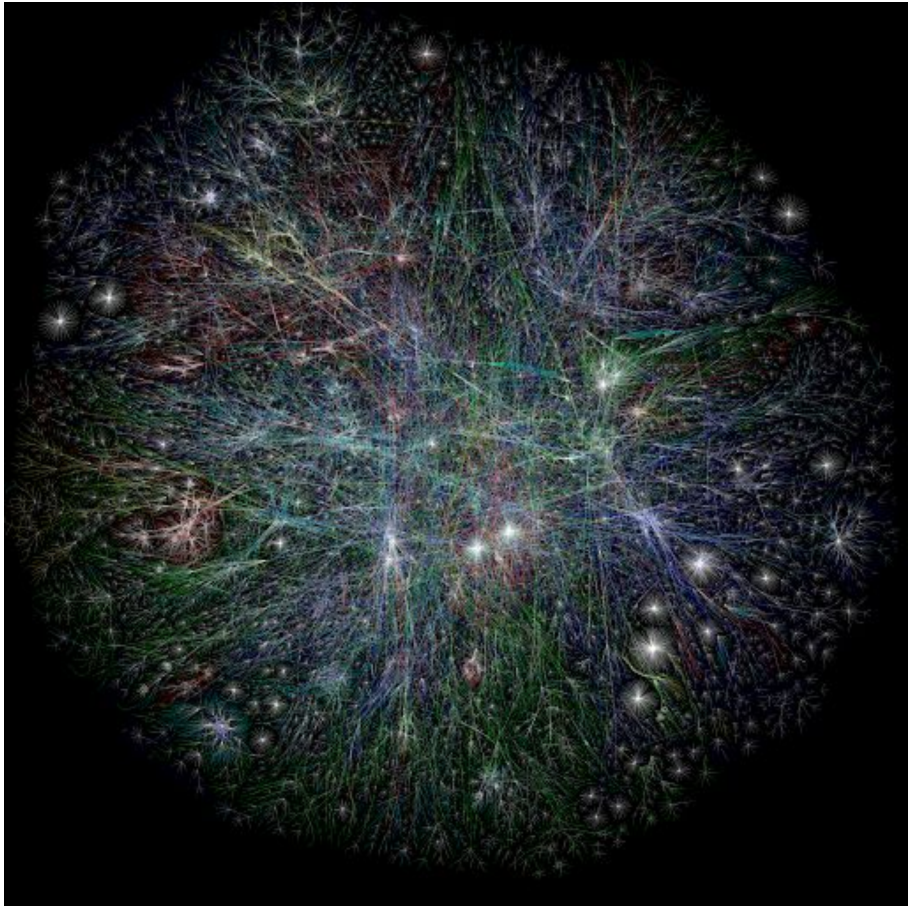
Internet Routing expert

Introduction

BGP is used to glue it all together.

RPKI is how we secure the Internet's routing system.

RPKI is a distributed cryptographically verifiable database.

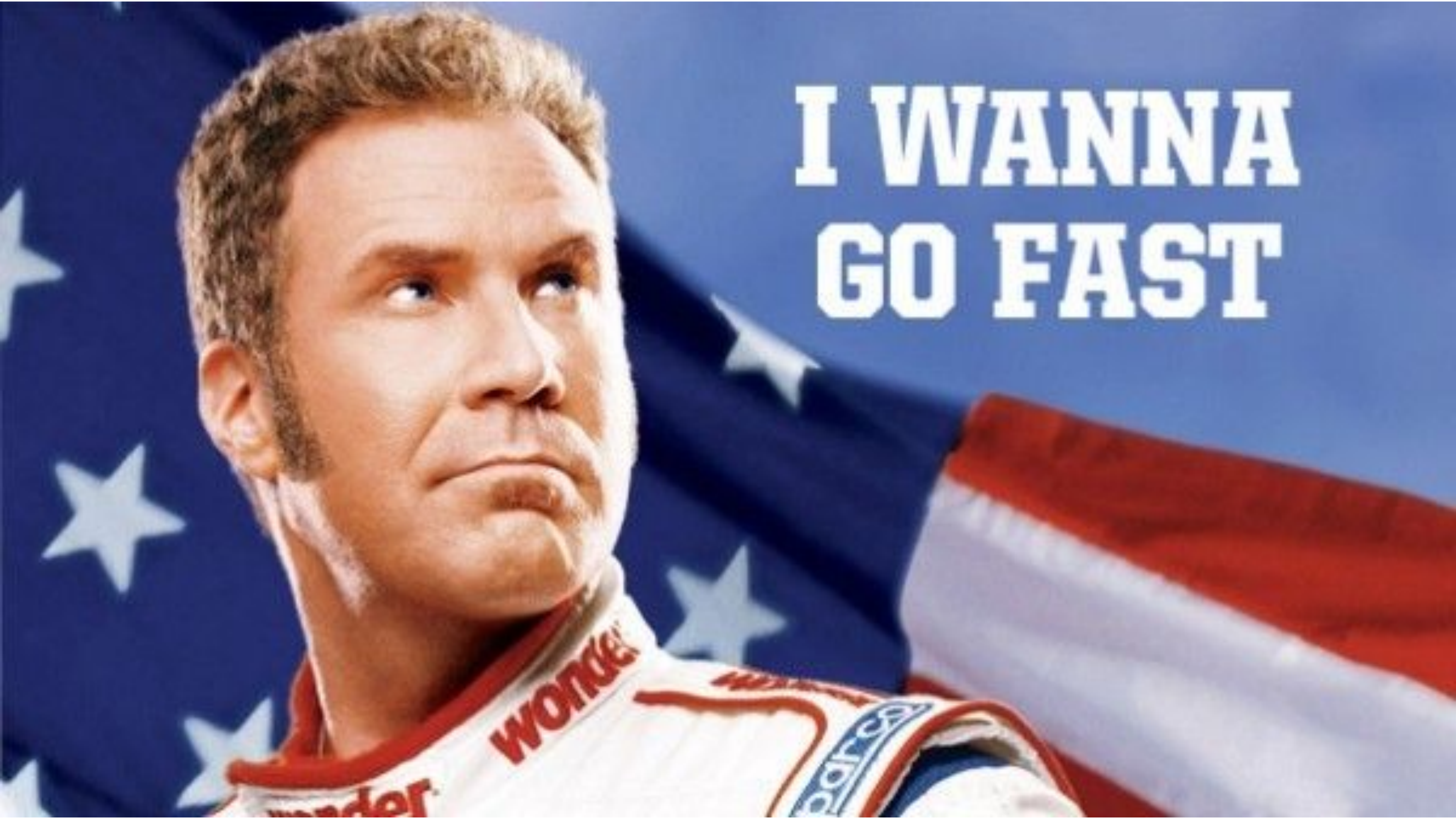


opte.org

Why any of this matters?

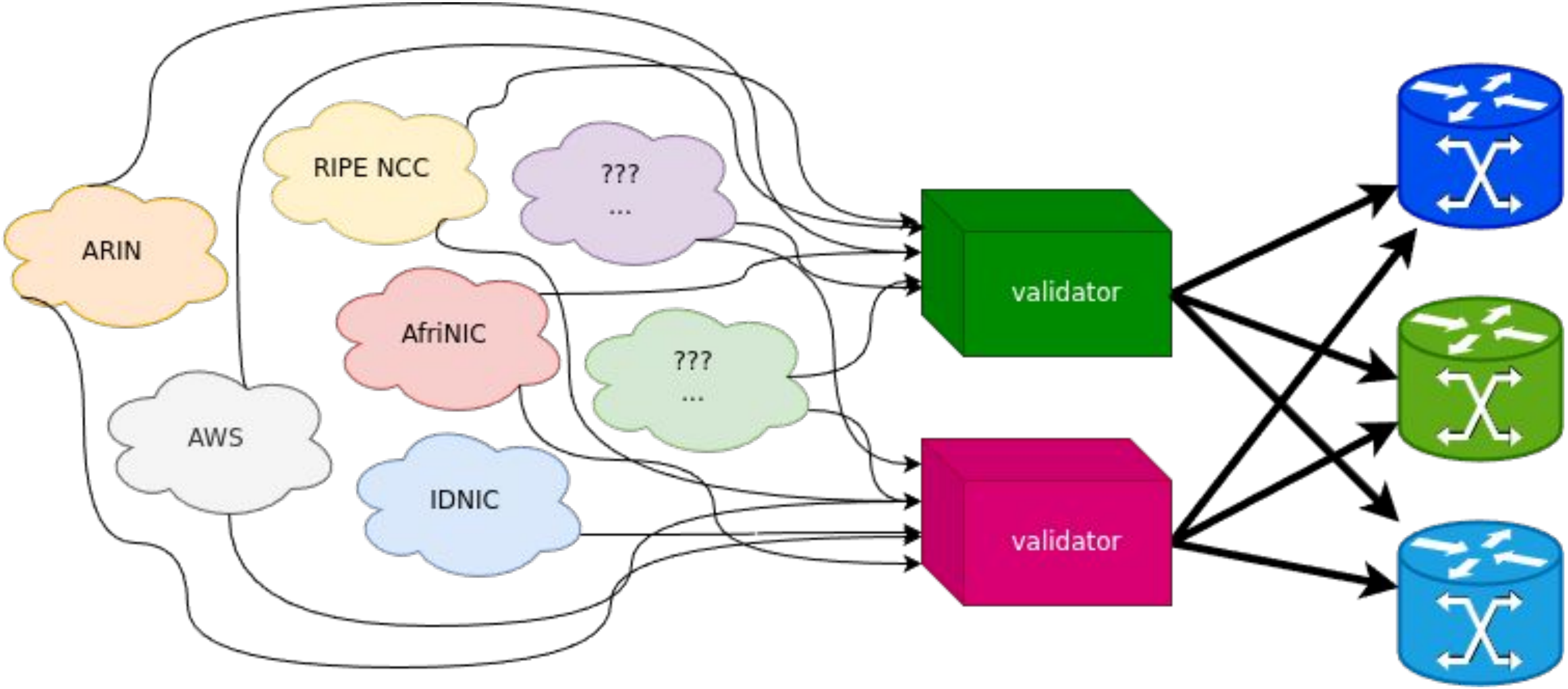
To make use of RPKI, the data needs to be distributed (obviously!)

It takes **between 10 and 60 minutes** for new RPKI to propagate: https://www.iijlab.net/en/members/romain/pdf/romain_pam23.pdf


I WANNA
GO FAST
wonder
wonder
sparco

High level overview of RPKI data supply chain

RPKI Repositories → *Rsync / RRDP* → Validators → BGP Routers



Delay factors:

Single-source…
Latency…
Congestion…
Certification…
Timers…

Protocol Efficiency

NRT
PEK
IAD
AMS
GRU
CPT

Job, why not just use cloud?

Proportions of the RPKI distribution problem space

Count: the RPKI database currently is ~ **500,000** tiny files

Churn: ~ half those files change at least once a day

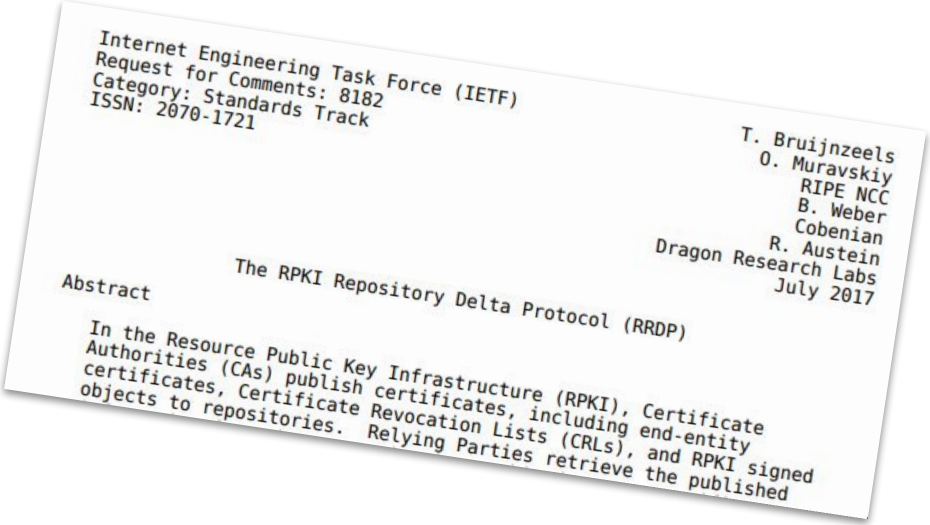
Total size: ~ 1.1 gigabytes

Consumers: ~ 5000 validators

Trend: up and to the right

https://labs.ripe.net/author/job_snijders/rpkis-2024-year-in-review/

So, what’s used to distribute RPKI data today?



Internet Engineering Task Force (IETF)
Request for Comments: 8182
Category: Standards Track
ISSN: 2070-1721

T. Bruijnzeels
O. Muravskiy
RIPE NCC
B. Weber
Cobenian
R. Austein
Dragon Research Labs
July 2017

The RPKI Repository Delta Protocol (RRDP)

Abstract

In the Resource Public Key Infrastructure (RPKI), Certificate Authorities (CAs) publish certificates, including end-entity certificates, Certificate Revocation Lists (CRLs), and RPKI signed objects to repositories. Relying Parties retrieve the published

But… rsync is efficient, right?!

Yes and no!

Rsync is efficient because it transfers “only the difference”...

But, calculating the difference
also consumes CPU & network resources!

Measuring Rsync

Interval	Bandwidth consumption
Minimum	~ 4 megabytes
Rsync every 15 minutes	~ 40 megabytes
Rsync every 60 minutes	~ 52 megabytes

Surely, RRDP was an improvement?

RRDP in a nutshell

- HTTP-based protocol, static pre-calculated content
- All add/update/delete operations are written into a “journal”
- Validators download the “journal” and replay it

Consequence:

Download of lots of data already overtaken by events

Measuring RRDP

Interval	Bandwidth consumption
Minimum	~ 0.5 megabytes
RRDP every 15 minutes	~ 5 megabytes
RRDP every 60 minutes	~ 100 megabytes

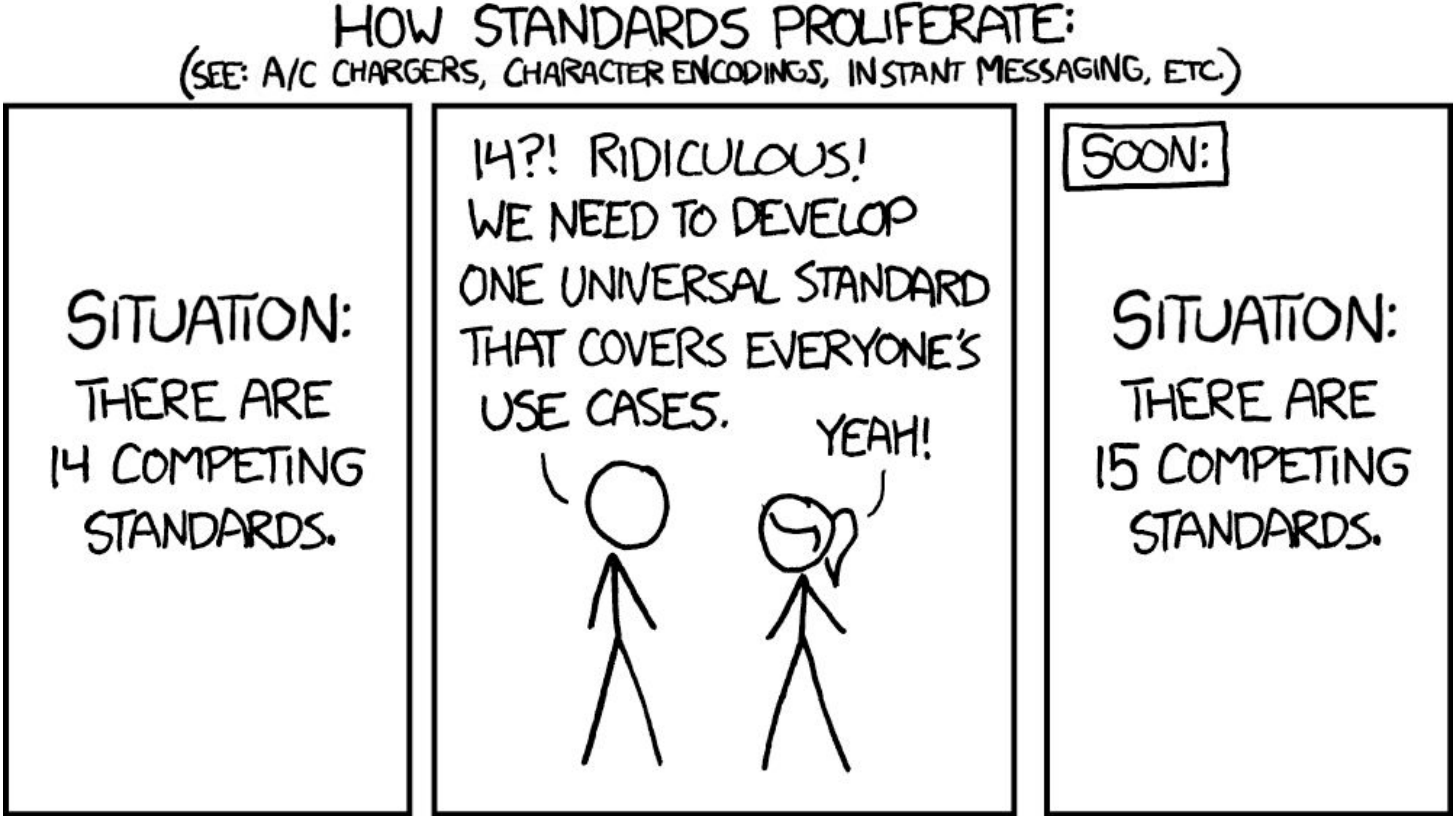
Comparing Rsync and RRDP

Interval	Rsync	RRDP
Minimum	~ 4 megabytes	~ 0.5 MB
Every 15 minutes	~ 40 megabytes	~ 5 MB
Every 60 minutes	~ 50 megabytes	~ 100 MB

Design issues

- Rsync servers vulnerable to denial of service (CPU hogging)
- RRDP servers can DoS clients (disk space hogging)
- RRDP “loss of state” → reinitialize through full download
- RRDP desynchronization can cause “Thundering herds”
- A slow server slows down parts of the system

Neither protocol really is optimal


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

What *is* the Erik Synchronization Protocol?

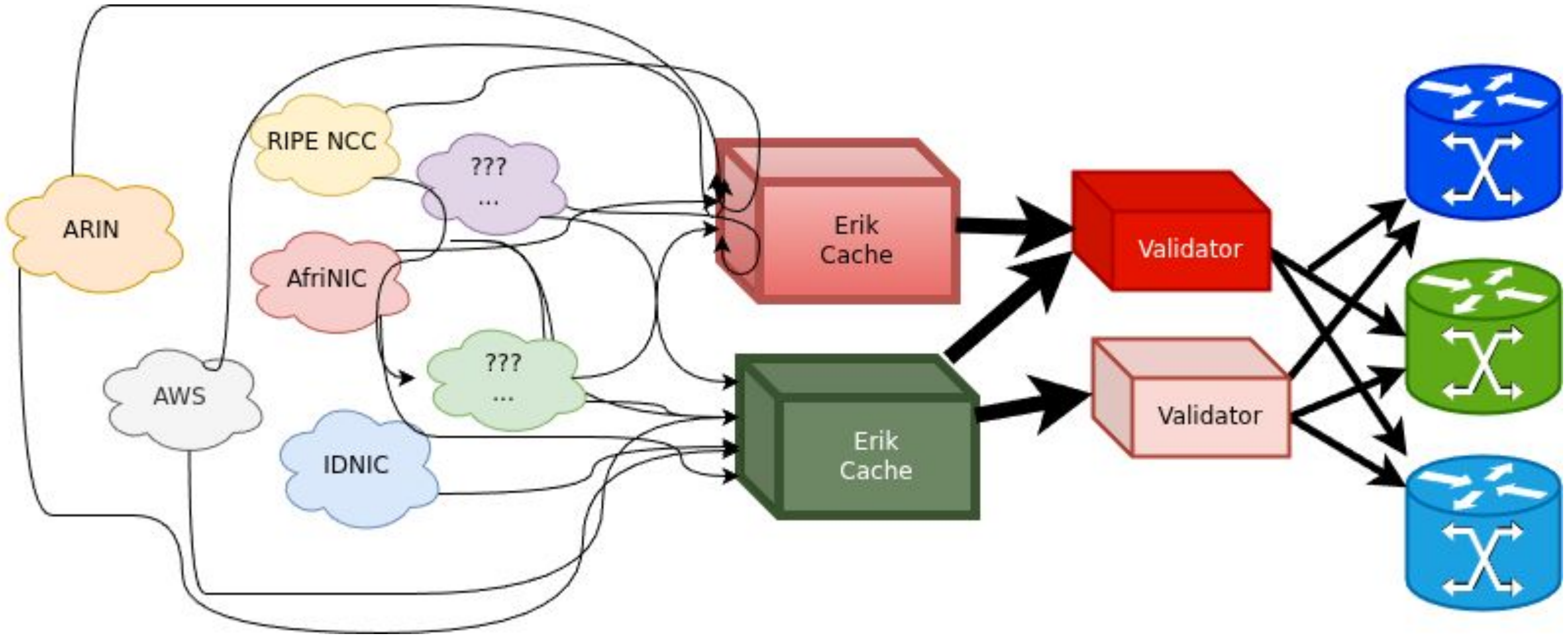
A data replication system using the following components:

- Merkle Trees (1979)
- Content-addressable naming scheme (1950s?)
- Timestamp-based concurrency control (1975)
- HTTP transport (1989)

Named in honor of Erik Bais who passed away in 2024.

Erik Protocol architecture: CDNify “everyone”

Repositories → *Rsync / RRDP* → **Erik** → Validators → BGP Routers



Advantages of the Erik Synchronization Protocol

- Clients “jump to latest” (like Rsync)
- Download “only what changed” (like Rsync)
- Static pre-calculated content (like RRDP)
- HTTP-based (like RRDP)
- Light on state, no “session” (like Rsync, unlike RRDP)
- Easy to combine with existing protocols (like Rsync, unlike RRDP)
- EFFICIENT
- FAST
- CHEAP

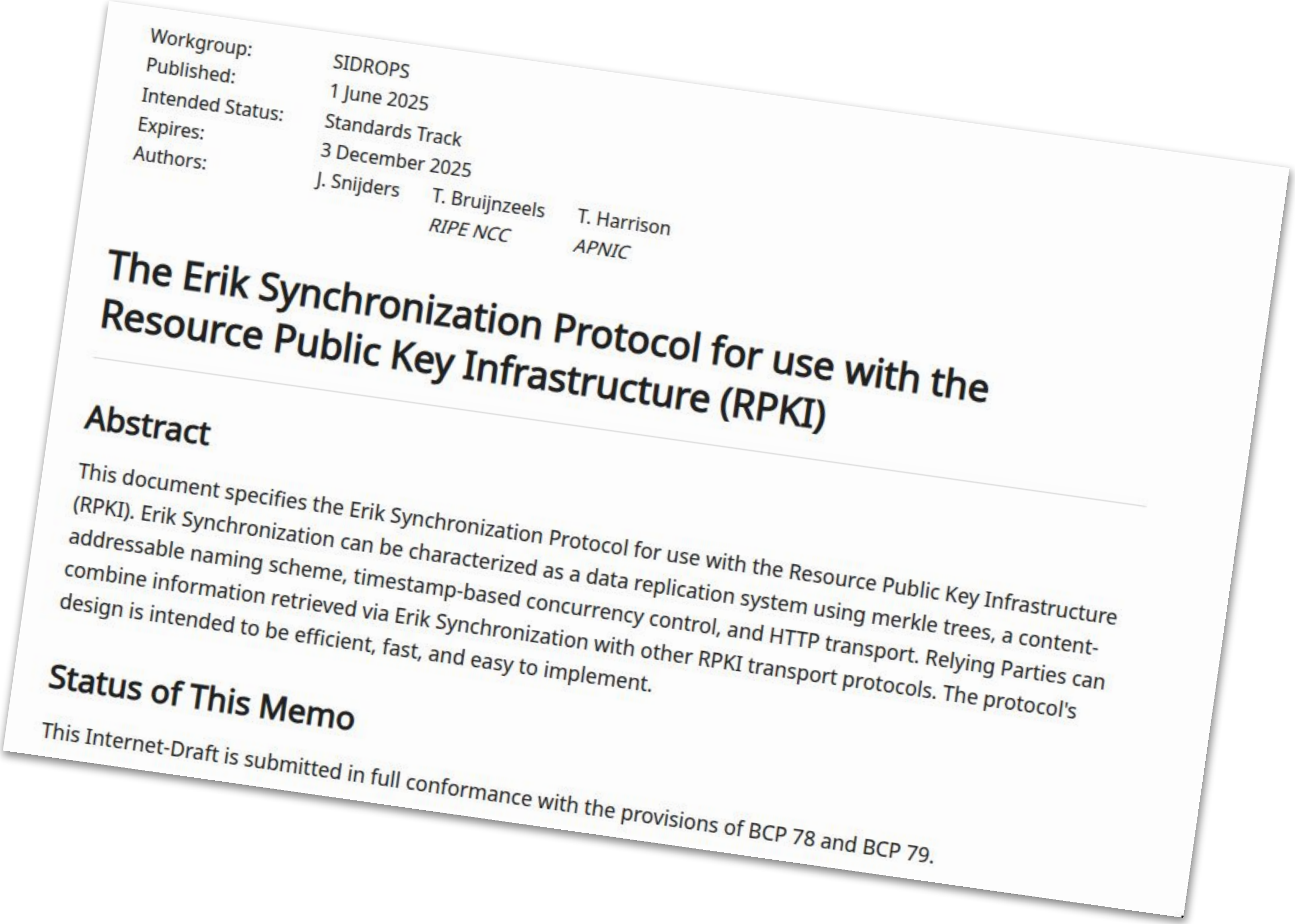
Erik servers are accelerators!

Comparing Rsync and RRDP and **Erik**

Interval	Rsync	RRDP	Erik
Minimum	~ 4 megabytes	~ 0.5 MB	~ 0.5 MB
Every 15 minutes	~ 40 megabytes	~ 5 MB	~ 4 MB
Every 60 minutes	~ 50 megabytes	~ 100 MB	~ 20 MB

Todo list

- Write IETF draft
- Write software
- Measure
- Experiment
- Optimize
- Iterate
- Deploy



Workgroup:	SIDROPS
Published:	1 June 2025
Intended Status:	Standards Track
Expires:	3 December 2025
Authors:	J. Snijders T. Bruijnzeels *RIPE NCC* T. Harrison *APNIC*

The Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI)

Abstract

This document specifies the Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI). Erik Synchronization can be characterized as a data replication system using merkle trees, a content-addressable naming scheme, timestamp-based concurrency control, and HTTP transport. Relying Parties can combine information retrieved via Erik Synchronization with other RPKI transport protocols. The protocol's design is intended to be efficient, fast, and easy to implement.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

How can you help?

Sponsor me!

Help pay for my food & mortgage https://sobornost.net/~job/

Review the specification https://github.com/job/draft-rpki-erik-protocol/

Donate compute / network resources for research! HORSEPOWER!!!!

QUESTION
THE
ANSWERS