

~~deprecate-rsync~~

prefer-rrdp

(draft-ietf-sidrops-prefer-rrdp-01)

first objective

Promote RRDP to a mandatory to implement, and preferred repository access method, so that the operational dependence on rsync infrastructure is reduced

phases

0

- If supported, RRDP **MUST** be made highly available (update 6481)
- RRDP unavailable, Relying Parties **MUST** fall back to rsync (update 8182)

1

- Repositories **MUST** support RRDP as a high availability service (update 6481)
- RRDP unavailable, Relying Parties **MUST** fall back to rsync (update 8182)

2

- Relying Parties **MUST** support and prefer RRDP. (update 8182)
- RRDP unavailable, Relying Parties **MUST** fall back to rsync (update 8182)

➡ Fall back strategies? Retries? Avoid a thundering herd?

➡ Phase 0: RPs **MUST** prefer RRDP if they support it and it is available?

long term objective

Remove the operational dependency on rsync:

- Simplify Relying Party code
- Simplify Repository operations

Requires:

- Operational experience and measurements
- A plan for rsync URIs

rsync names as identifiers

In order to remove rsync URIs, we need a lot of changes:

- RPKI Certificate profile (6487)
- Out-of-band protocol (8183)
- Publication Protocol (8181), and RRDP (8182)
- Provisioning Protocol? (6492)

proposal

Focus on prefer RRDP for now.

Spin off new work for having transport agnostic names.

prefer RRDP status

RRDP is supported by 4 out of 5 RIRs, the remaining plans to deploy very soon (weeks?).

RRDP is supported by all delegated RPKI software. One upgrade is pending.

RRDP is supported by 6 out of 7 Relying Party implementations, and is under development for the 7th.

➡ Aim for publication for phase 2? End of 2021?