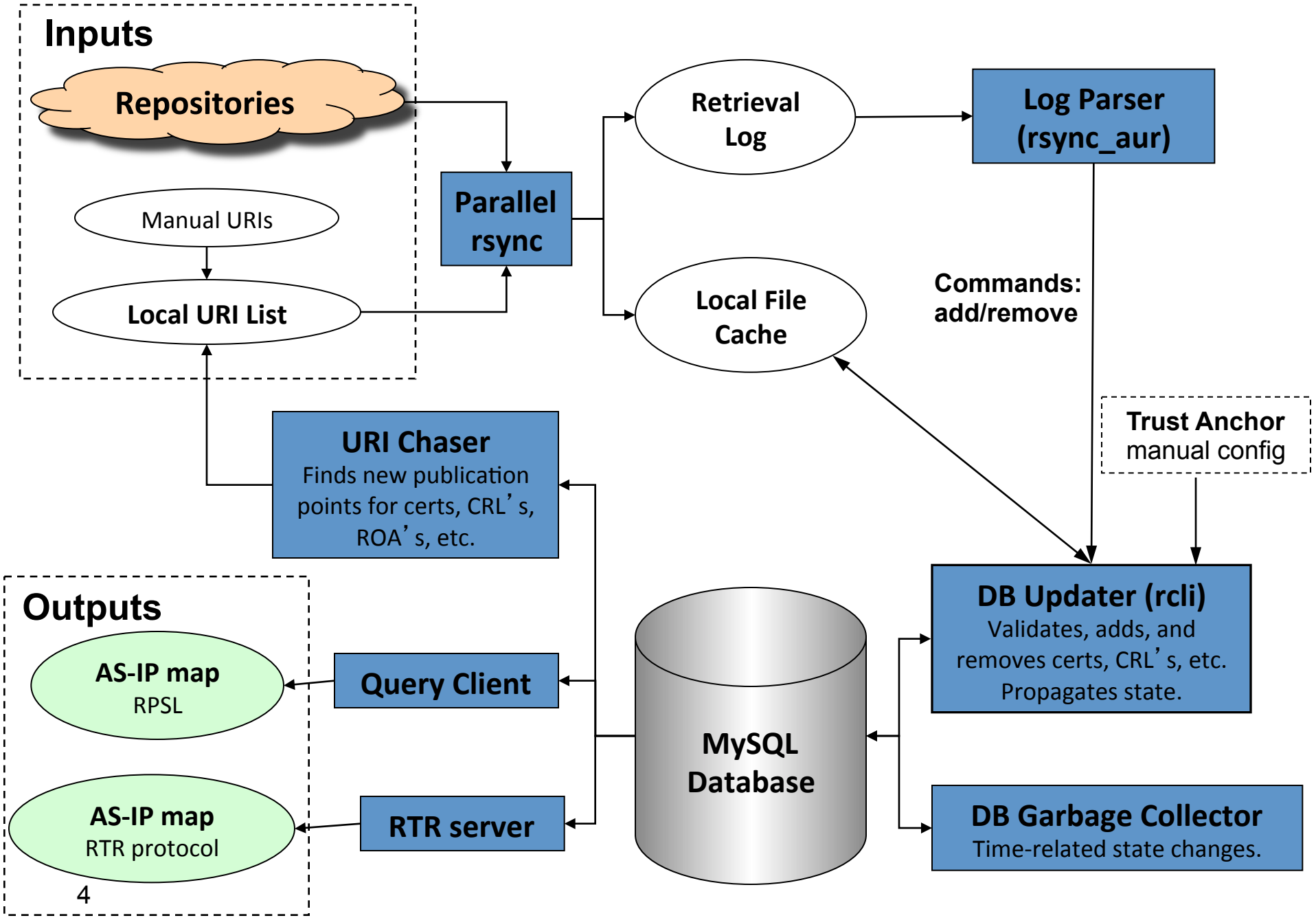
BBN Relying Party Software for the RPKI

BBN RP Software Key Features (1/2)

- Intended to be product-quality implementation of validator + RTR.
- Designed to be fast.
  - Local database of certificates, CRLs, and RPKI signed objects provides efficient access to verified, immutable object values.
  - Each file in the local repository (cache) is read and parsed only once.
  - Written in C, and a very small amount of Python.
- Designed to be flexible. Incremental, deferred validation of signed objects, tailored so that these objects can arrive in any order.
  - Database maintains state so that expensive operations like signature verification and hash computation are performed only once.
  - Validation usually proceeds in a top-down manner from the trust anchors. However, data can be efficiently pre-loaded in an "unknown" validation state even if parent certificates are not yet available.

BBN RP Software Key Features (2/2)

- Standards compliant (SIDR and PKIX)
  - Uses OpenSSL, Cryptlib, and BBN ASN.1 library in effort to enforce strict compliance with standards.
  - Open source, multi-platform: Linux, FreeBSD, NetBSD and OpenBSD, more later, incl. OS X)
- Designed to be robust.
  - Parallel rsync avoids DoS by slow repository. Bonus: performance enhancement for very flat repositories.
  - Local trust anchor support [draft-ietf-sidr-ltamgmt]
  - Performed DoS assessment


Inputs
Repositories
Manual URIs
Local URI List
Parallel rsync
Retrieval Log
Log Parser (rsync_aur)
Local File Cache
Commands: add/remove
Trust Anchor
manual config
URI Chaser
Finds new publication points for certs, CRL's, ROA's, etc.
Outputs
AS-IP map
RPSL
Query Client
AS-IP map
RTR protocol
RTR server
MySQL Database
DB Updater (rcli)
Validates, adds, and removes certs, CRL's, etc.
Propagates state.
DB Garbage Collector
Time-related state changes.

System Performance

- Tested the system using simulated repository data generated from RIR "profiles".
  - 9,932 CA certificates
    13,292 EE (embedded) certificates
    6,646 CRLs
    6,646 ROAs
    6,646 manifests
    43,162 objects in all => 47 minutes 26 seconds

  Note that these numbers are old.
- Whole Internet deployment (~300,000 objects) => ~ 5.6 hours (one time cost of initializing DB)
- Typical daily update (3,000 objects) => less than five minutes
- We're working to improve these numbers

Planned Software Enhancements

- Ghostbusters
- Update RTR
- Extensive testing
  - Repository generator has been designed to simulate various levels of misbehavior.
- Improve performance, fix bugs, …