

fast-repo update

5/29/2019

original motivation

- slide from 10/10/2018:

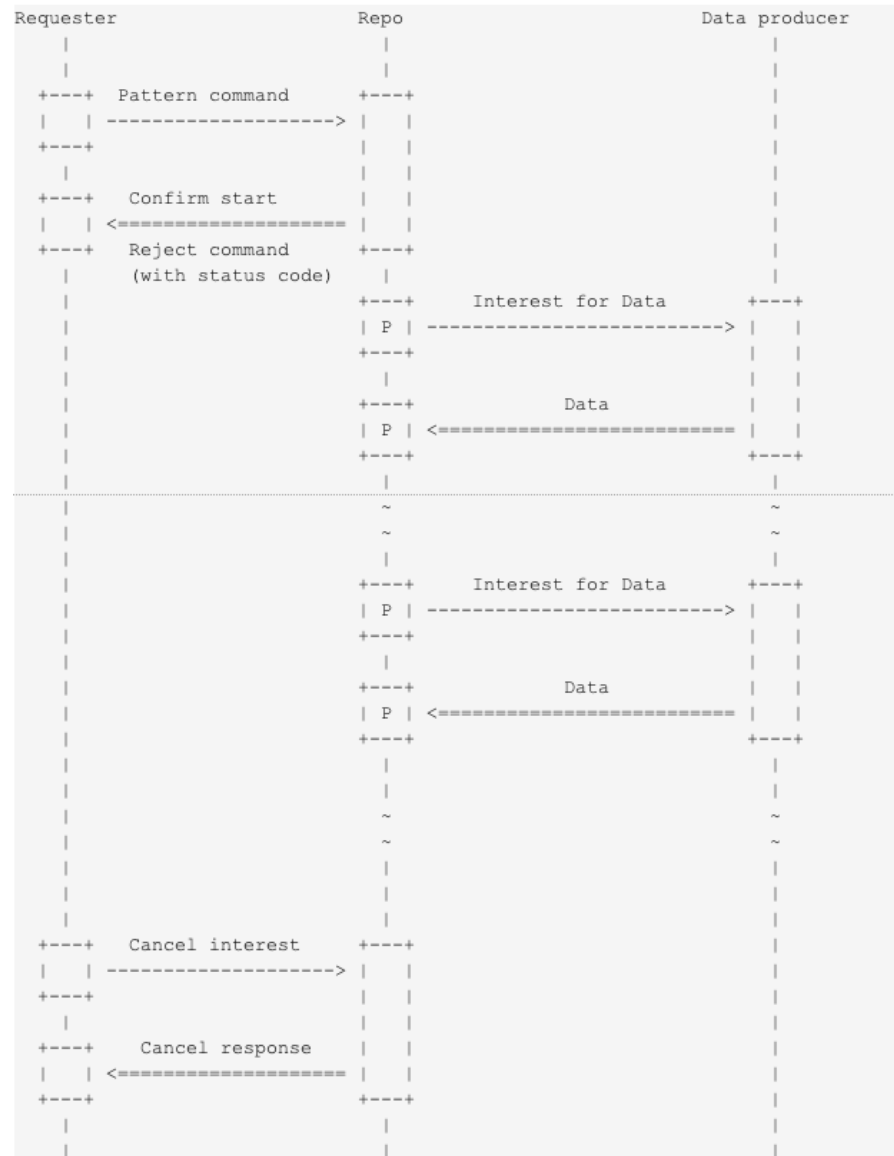
what is needed

- store generalized objects in persistent storage
- store generalized object **streams** in persistent storage
- store ndnrtc **streams** in persistent storage
- serve stored data **immediately**

how it was approached

- GObj, GObjStream and NDN-RTC are application-defined namespaces
 - => requires custom fetching code
- fast-repo implements **pattern-fetching**
 - if fast-repo **supports** requested pattern, it'll fetch the custom namespace
 - patterns are implemented as subclasses in the codebase
 - right now, we only have NDN-RTC v3 pattern implemented

















pattern fetching protocol ([by Xinyu](#))






pattern fetching protocol (cont.)

- pattern command:
 - /repo/pattern/<RepoCommandParameter>/<timestamp>/<random-value>/<SignatureInfo>/<SignatureValue>
- **Name** (required) should be /<PatternName>/<FetchPrefix>. “ndnrtc3” “/ndn/edu/ucla/alex/ndnrtc-stream”
- **PatternName** is the name of FetchPattern.
- **FetchPrefix** is the prefix which will be fetched.

fast-repo vs repo-ng

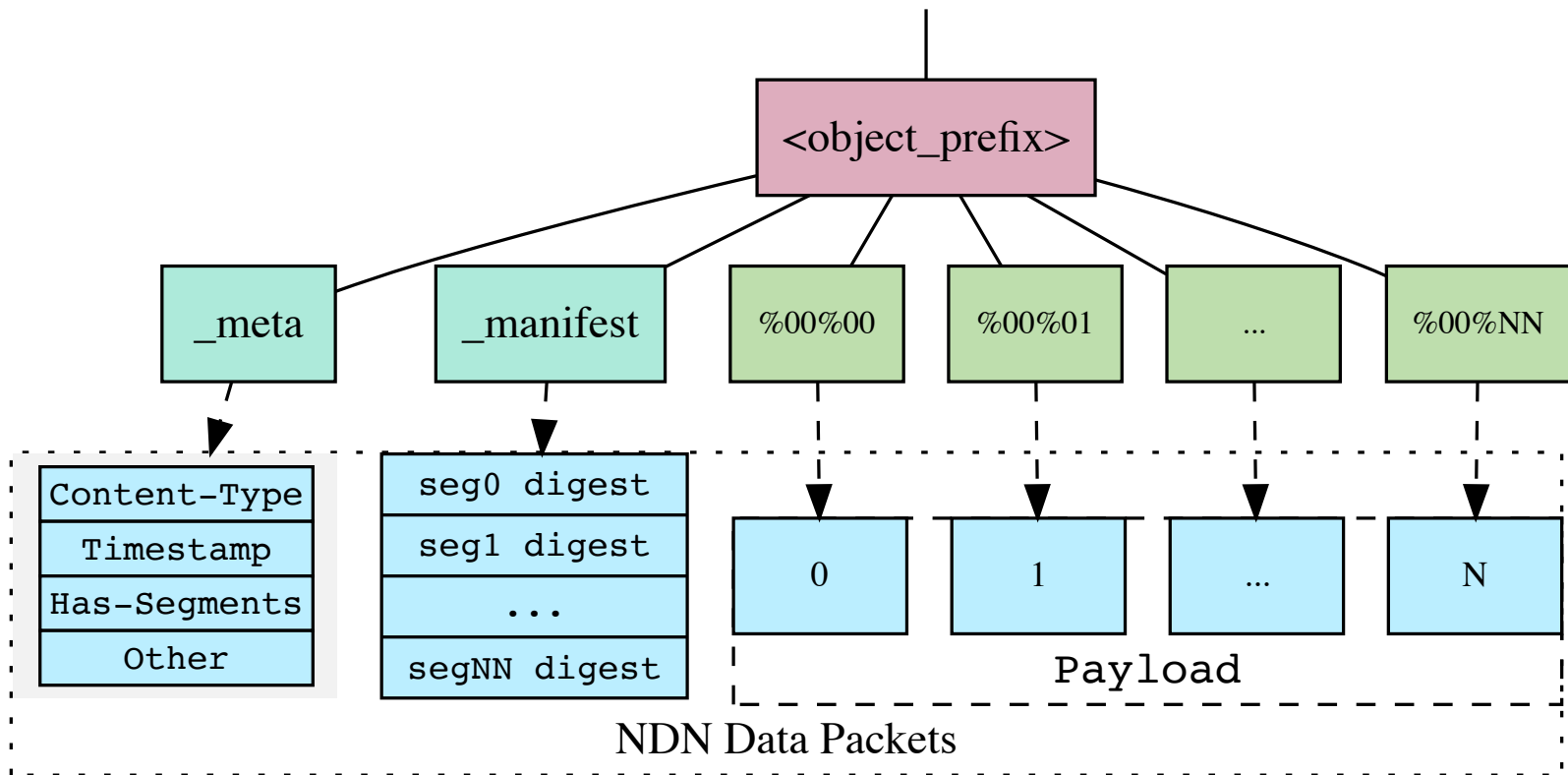
fast-repo	repo-ng
<i>RocksDB</i>	<i>SQL DB (SQLite?)</i>
      	        

-  - won't implement
-  - TBD
-  - implemented

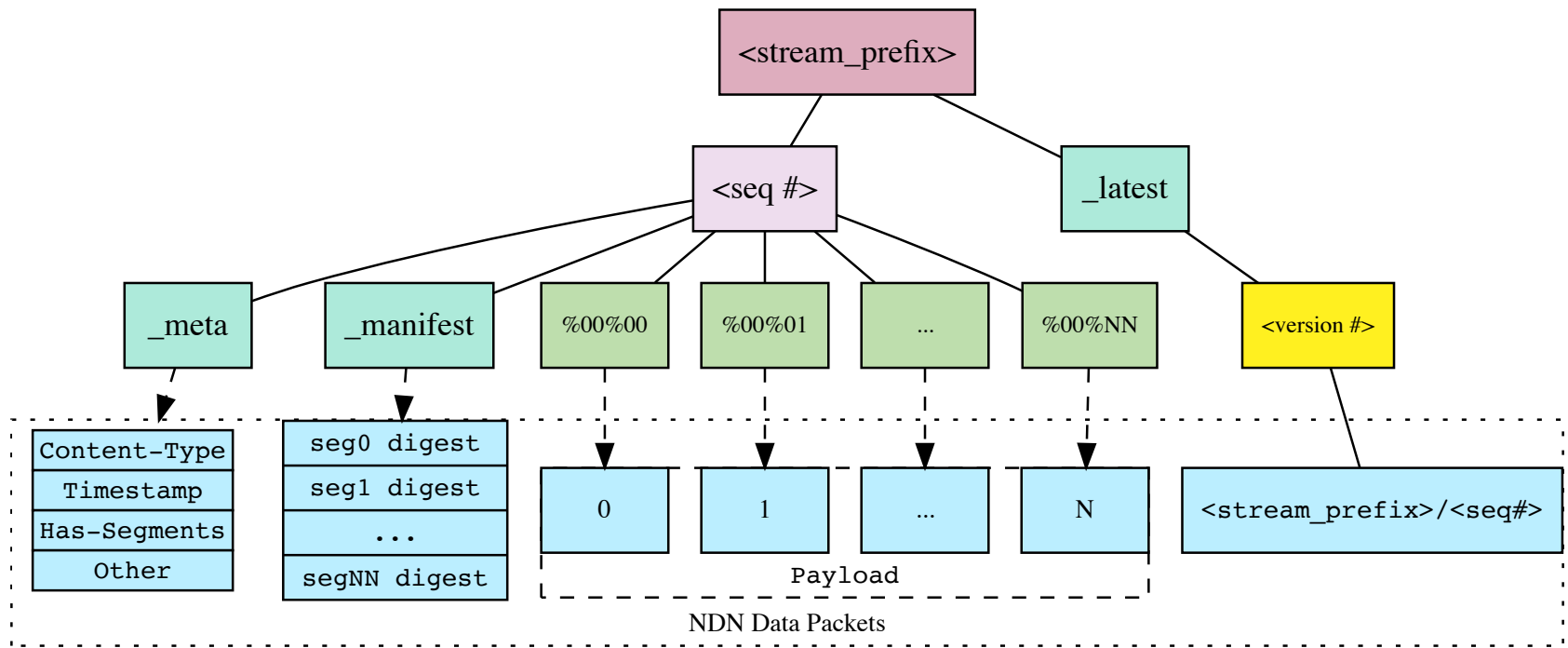
fast-repo requirements

- <https://github.com/remap/fast-repo>
- C++ codebase is set up with the following dependencies:
 - NDN-CPP
 - NDN-CNL
 - Protobuf
 - RocksDB
 - Boost

GObj namespace



GObjStream namespace



NDN-RTC v4 namespace

