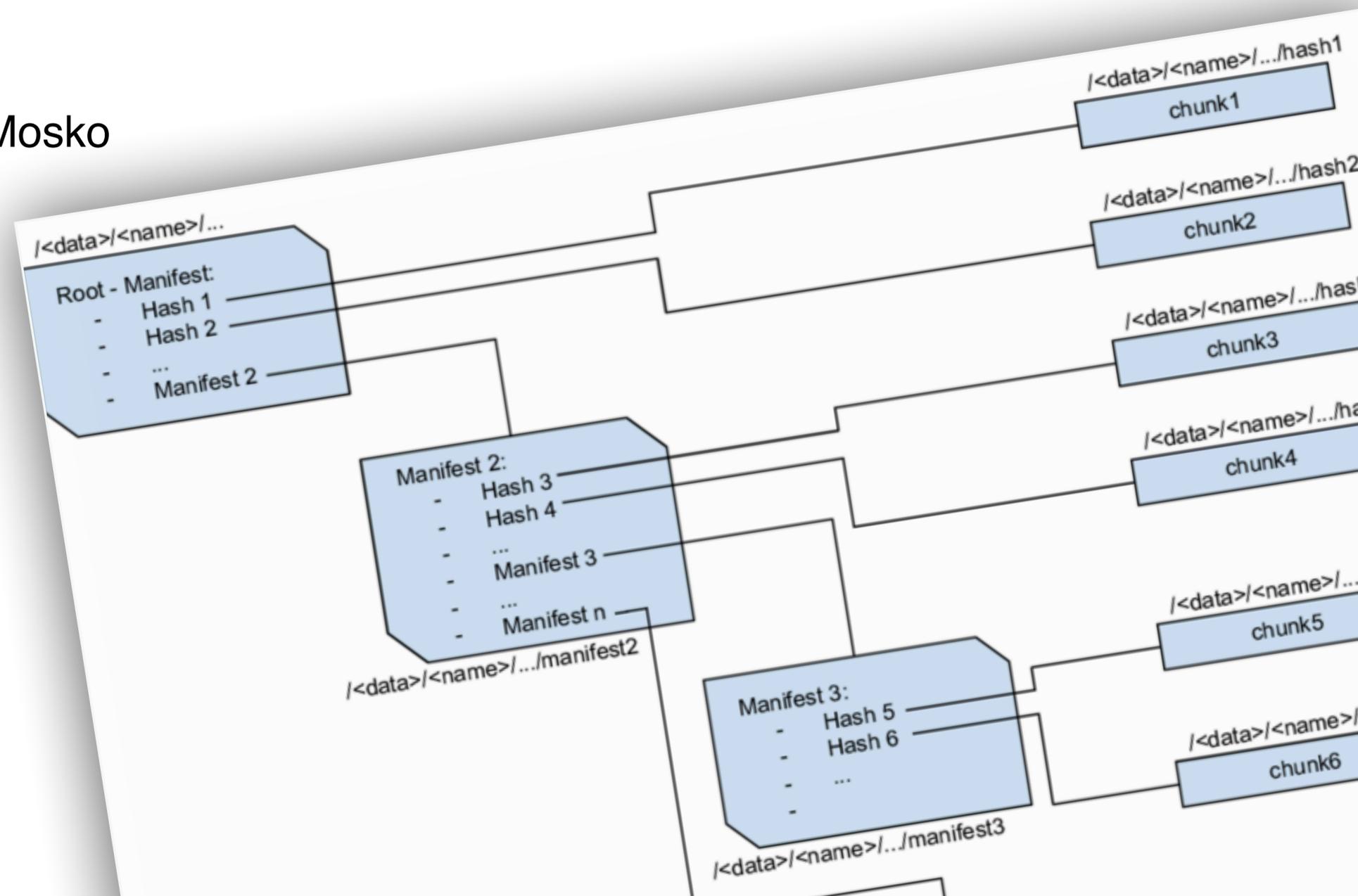


Recap of Recent FLIC Discussions (FLIC = File-Like ICN Collections)

Christian Tschudin, U of Basel, Switzerland
based on input from Dave Oran and Marc Mosko

ICNRG Online Meeting
2020-12-01, 15:00 to 18:00 UTC



FLIC Timeline

- 00-draft June **2017**: initial version by Christian Tschudin and Chris Wood
 - insights from (internal) C programming and CCN-light
- 02-draft Nov **2019**: Marc Mosko and Dave Oran joined
 - interest to join because of “manifests” (Oran) and “nameless objects” in CCNx (Mosko)
- **Spring 2020**: Marc Mosko writes a Python prototype, “namespace” concept (see links at the end of this slide set)
- **August 2020**:
 - comments by Cenk Gündoğa, Ken Calvert, ping-pong with Dave, Marc, Christian
 - “abstracting” the core of FLIC? “virtual blobs” by ChristianDiscussion is both about terminology and scope and complexity. Sample:


```
>>> FLIC describes a single file. No. By definition, FLIC must always describe two files: the manifest and the data.
```

Dave's assessment

- still some disagreement on how basic to keep the spec for its first version
- general agreement on registries for defining extensions with some details needing to be fully fleshed out
- general agreement that we need the “namespaces” stuff that Marc put together, but still to be decided if they only go in root manifests.
- agreement on need for some meta-data capability (size hints, possibly fetch order hints) but no agreement on exactly which mechanism to enhance (hash group types or something general to apply to any hash group)
- discussion still on waiting for more complex metadata machinery like extra manifest-wide metadata TLVs, but perhaps include the extension registry for future-proofing.

Marc's assessment

most recent (and pending) discussion contribution Sep 2, 2020/ICNRG list:
“Options for FLIC NameConstructor” . On FLIC document status:

- [we seem to agree] that we need name constructors so a client knows how to fetch content
- We need some way to extend the manifest, i.e. some sort of metadata
- [at least we need a] link section that points to external metadata objects with a TBD format and keep the core manifest the simple hash groups
- That said, I don't think the current writeup with a basic hash group and an annotated hash group are all that complex.

Christian's detour

Did a deep dive into decentralized Hypercore (previously named DAT):

- *immutable data blocks, hash pointers, signed Merkel trees*
- *uses several nameless "hypercores" to build higher-level data structures*
- *example: build a file system with*
 - . *one hypercore for content ("block level")*
 - . *one hypercore for the name bindings (directories, file attributes)*
- can inform FLIC scope, point at use cases beyond classic CCNx and NDN
- conceptual/structural insight: separate abstraction (hypercore) from their implementation (several files, e.g. separating signatures from data blocks)
- but also difference to FLIC: is there really a manifest? Only virtually?

FLIC continues

agreement among Dave, Marc and Christian to participate in more technical sessions

Links, from Marc's message Jul 31, 2020:

- updated FLIC spec <https://github.com/mmosko/icn-flic-manifest>
- update prepared for IETF 106
<https://github.com/mmosko/icn-flic-manifest/tree/master/docs>
- Python implementation of CCNx 1.0 (partial) with the new FLIC manifests
<https://github.com/mmosko/ccnpy>
- a FLIC group key proposal
<https://github.com/mmosko/ccnpy/blob/master/FLIC%20Group%20Key%20Proposal.docx>
- initial thoughts on FLIC key wrapping
https://github.com/mmosko/ccnpy/blob/master/flic_keywrap.md
- examples of manifest structures
https://github.com/mmosko/icn-flic-manifest/blob/master/FLIC_Update.md