# FLIC Manifests

### Goals

- Copy UNIX inode concept
  - Index tables and memory pointers
- Adaption to ICN
  - Hash values instead of block numbers
- Advantages
  - Single root manifest signature that covers all children nodes
  - No need for explicit chunking
  - Supports block-level de-duplification

### Limitations

All data leaves must be present at creation time

## FLIC Overview

root manifest

```
optional name:
  /icn/name/of/this/flic
HashGroup:
  optional metadata:
   block size, total tree size,
                                          leaf
   overall digest, locator, etc.
                                       data
 hash-valued data pointer ---->
                                                 sub manifest
 hash-valued manifest pointer ----
optional additional HashGroups ..
optional signature
```

### FLIC EBN

```
HashGroup := MetaData? (DataPointer | ManifestPtr)+
DataPointer := HashValue
ManifestPtr := HashValue
HashValue := OCTET[32]

MetaData := Property*
Property := Locator | DataBlockSize | OverallDataSize |
OverallDataSHA256Digest | ...
```

ManifestMsg := Name? HashGroup+

### FLIC Trees

• Skewed (similar to a list)

```
DDDDDDM--> DDDDDDM--> ..... DDDDDDM--> DDDDDDD
```

- Balanced
  - application data is all at the bottom of the tree

### Use Cases

- Block-level reduplication
- Growing ICN collections
- Republishing under a new name
- Data chunks of variable size