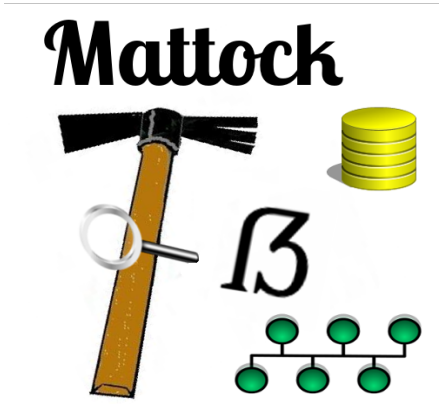
Todays schedule

- **Asynchronous processing & tool-chain approach**
- Integrity, privilege separation and capabilities.
- CarvFS & MinorFS
- MattockFS core design
- MattockFS as distributed-framework building block
- Installation (hands on)
- File-system as API (hands on)
- Python API (hands on)

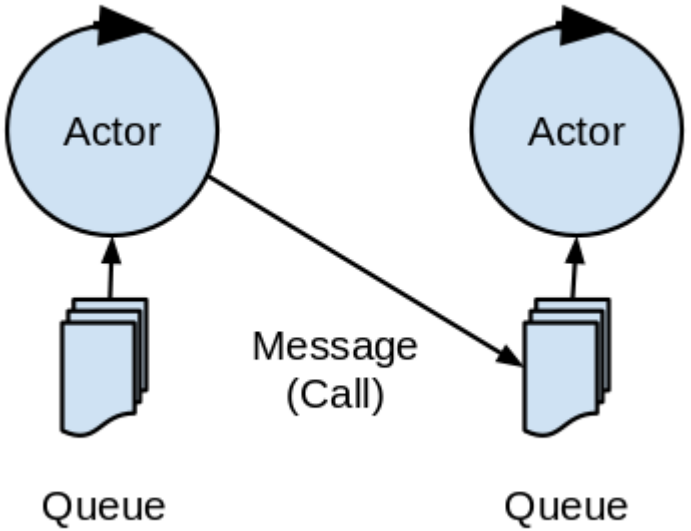


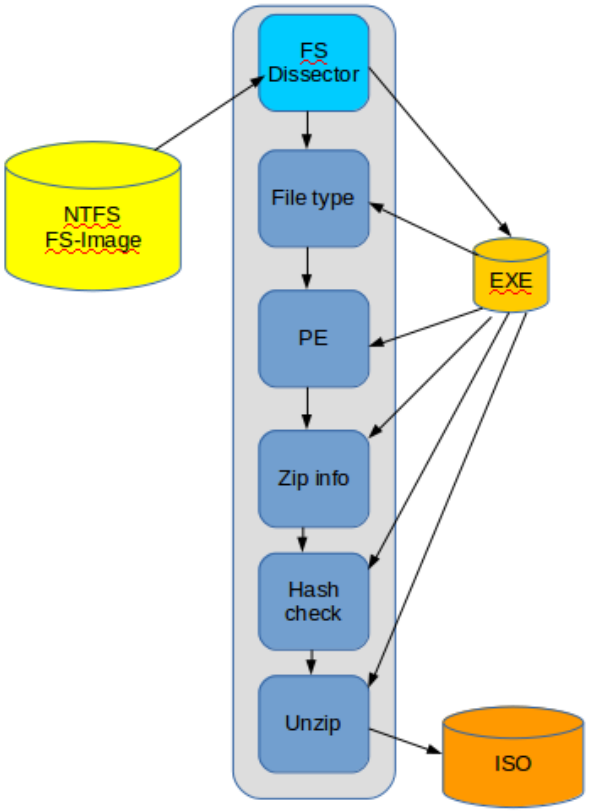
MattockFS

Computer-Forensics File-System

Asynchronous processing & the tool-chain approach.

Asynchronous processing and the tool-chain approach





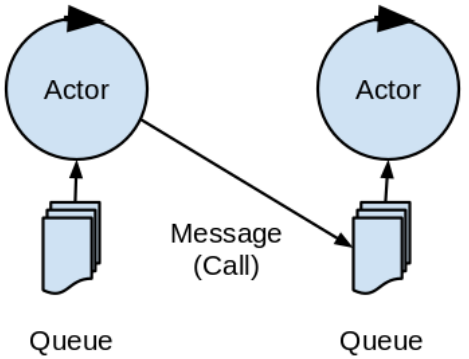
Base Concurrency Models

- Shared state concurrency
- Common address space (R+W)
- Locks & semaphores.
- Potential robustness issues.

- Message passing concurrency
- Private address space
- Queues
- Potential latency and queue size issues.

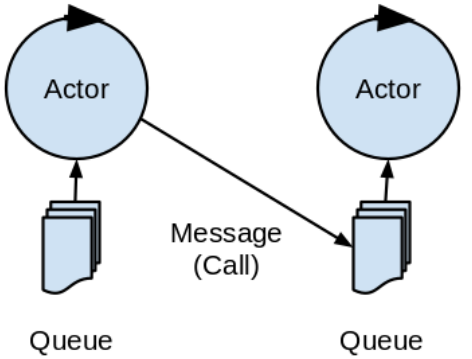
Actor model of computation

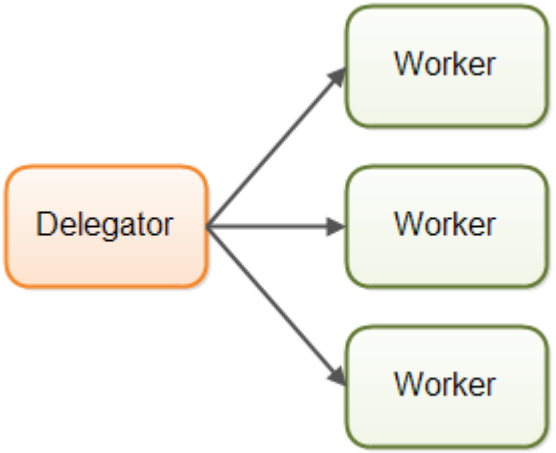
- Actors as universal fundamental primitives of concurrent digital computation.
- Event driven
- Communication by messages
- Fault tolerant by supervision

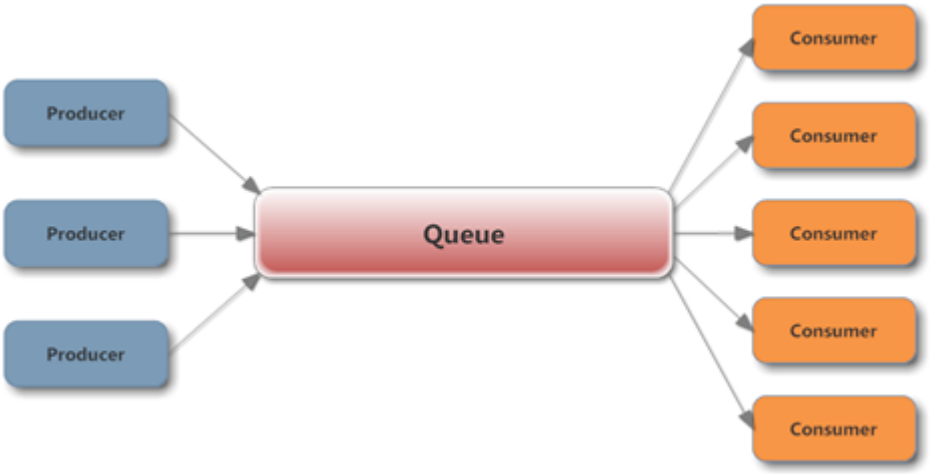


Message passing concurrency

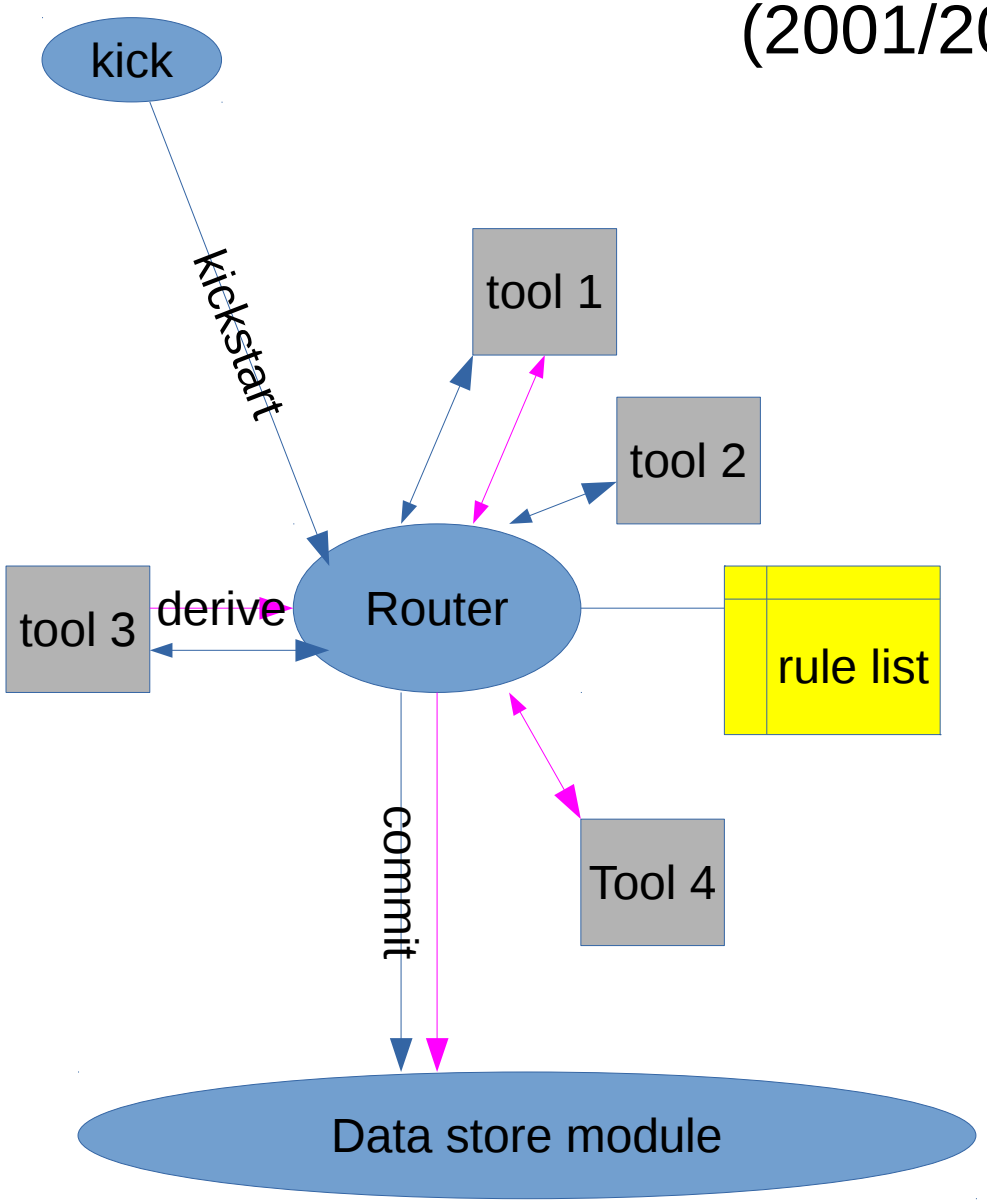
- Actors
- Workers
- Producers/Consumers





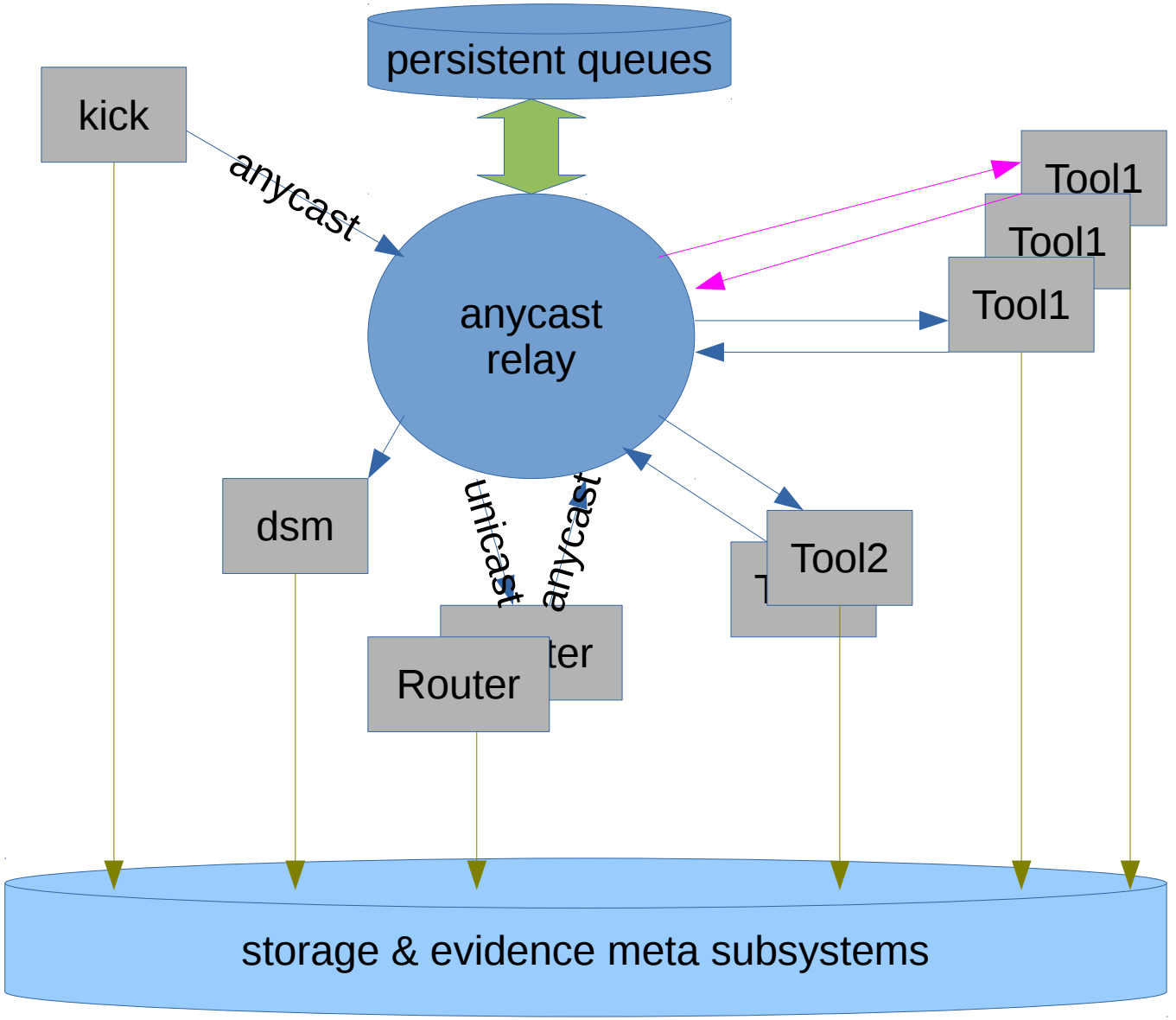


The Open Computer Forensics Architecture
(2001/2006-2012)



- Distributed
- Asynchronous
- Message passing concurrency
- Use existing tools/libs
- Fault isolation
- Recoverable failure
- Hundreds of disk images

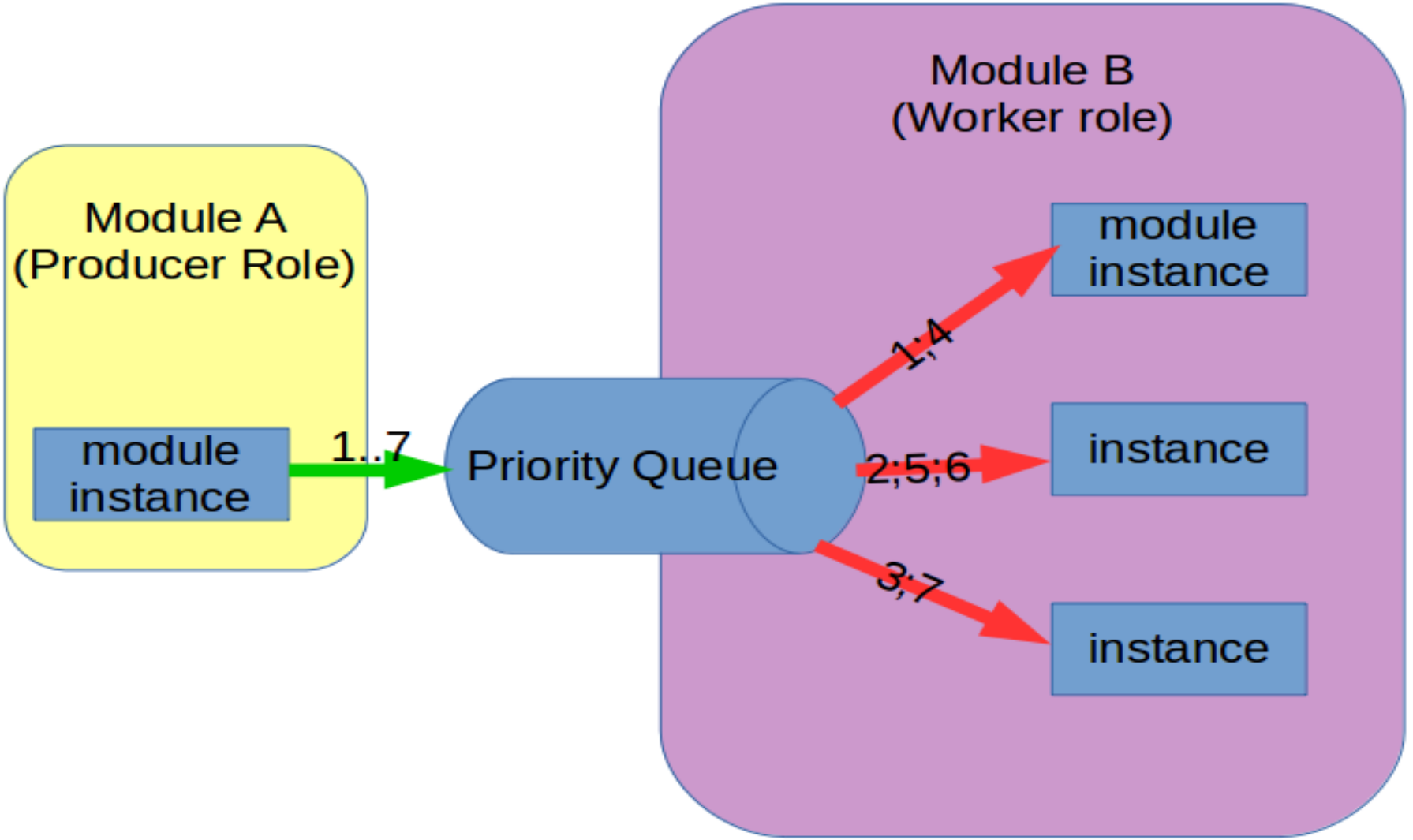
The Open Computer Forensics Architecture
(2001/2006-2012)



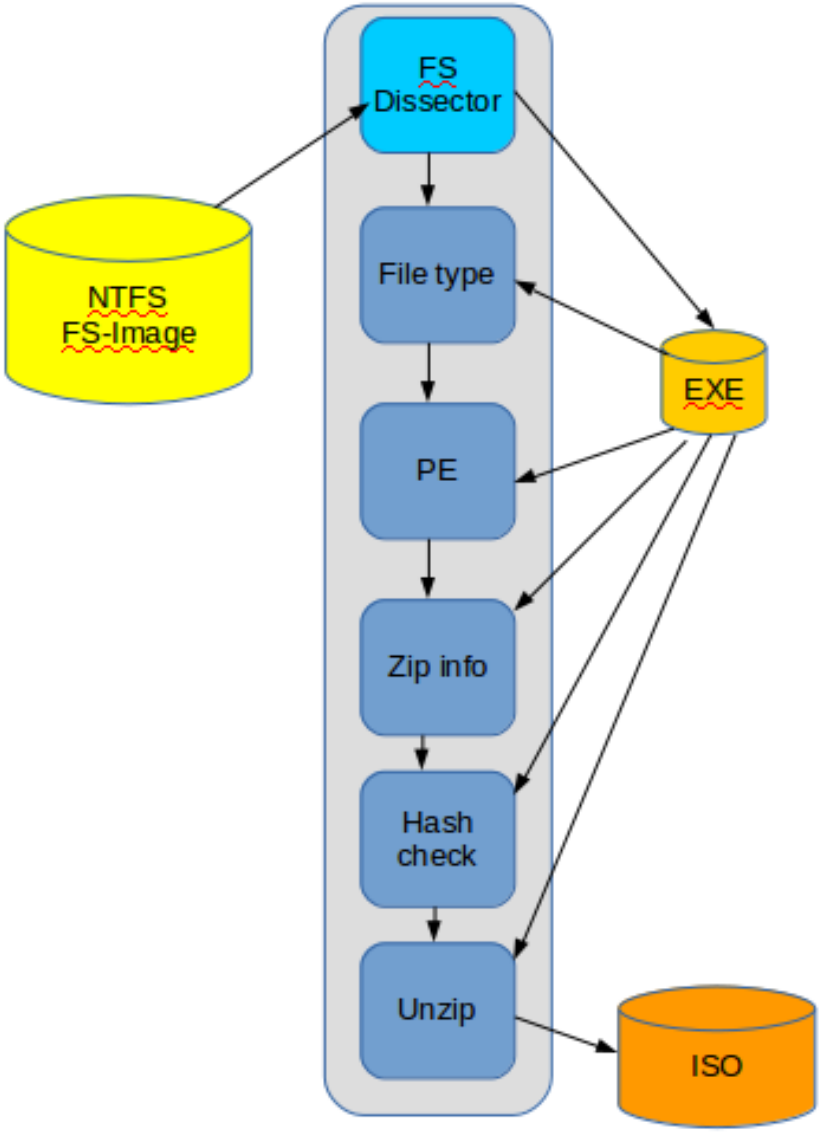
The OCFA Anycast

- Message bus solution
- Based on the **workers** concept
- Part of Open Computer forensics Architecture
- Persistent Priority Queues
- Virtually infinite size queues

Queues and workers

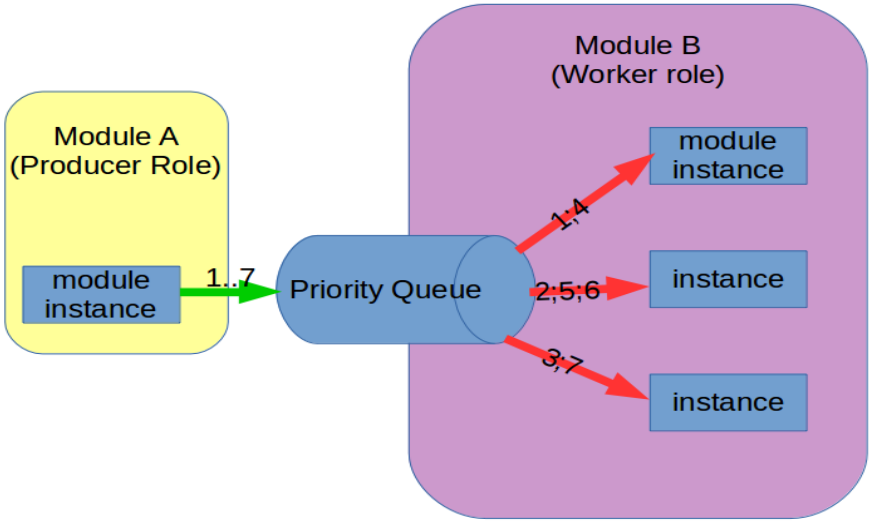


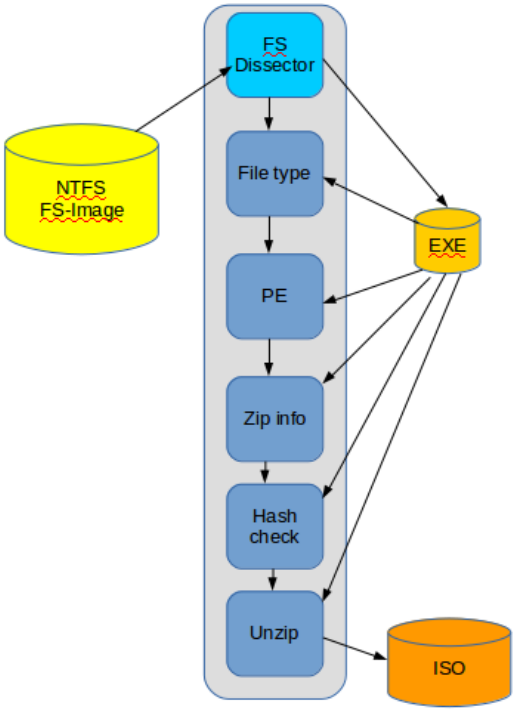
Example forensic tool-chain



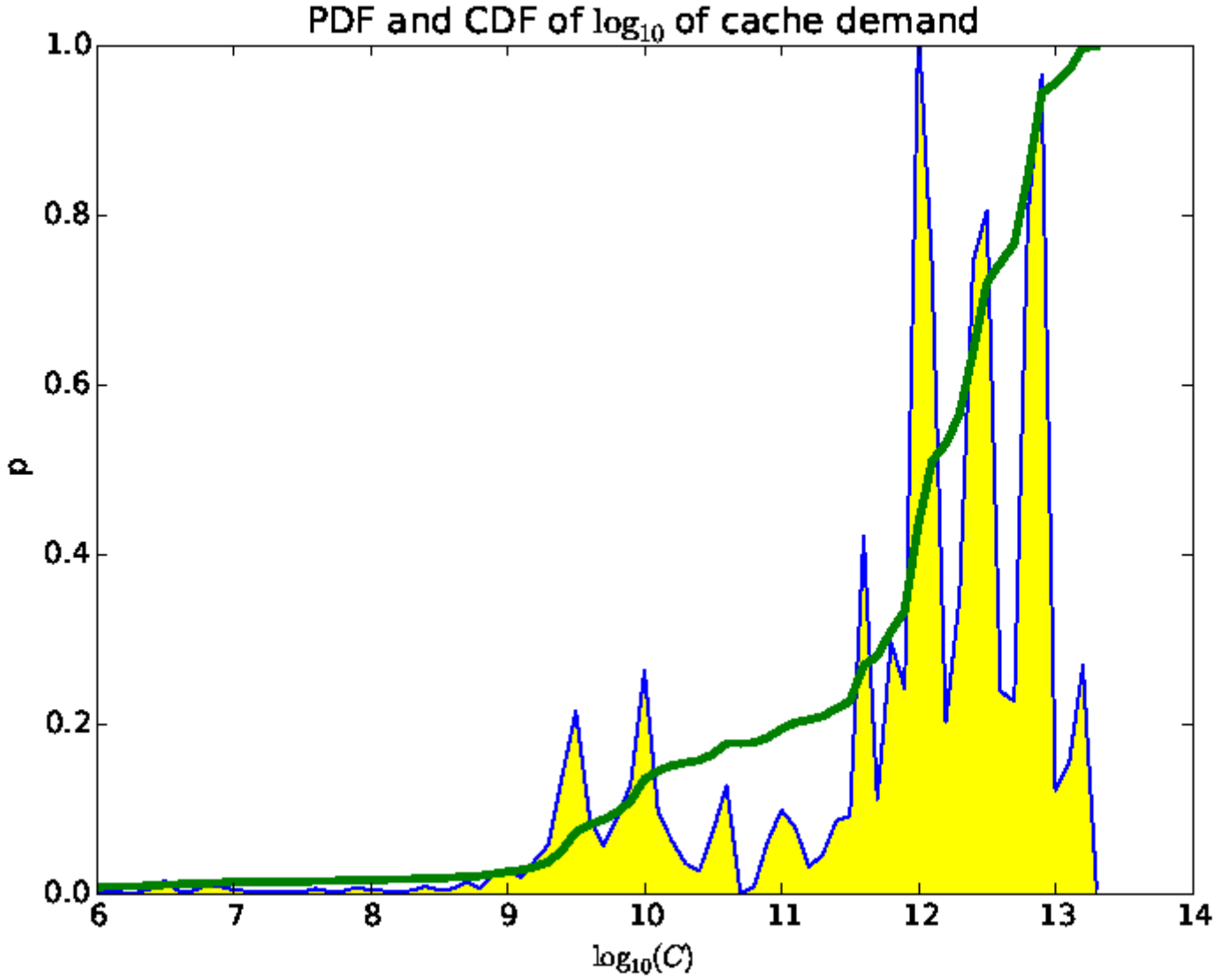
Issues with AnyCast

- Infinite size queues
- Multiple tools in tool-chain access same data
- Result: Many **page-cache misses**





Unthrottled asynchonous processing



Spurious reads in OCFA

- Page-cache misses
- Early hashing and Content Addressed Storage
- No locality of data design
- Server-side data entry

The Open Computer Forensics Architecture
(2001/2006-2012)

- Distributed
- Asynchronous
- Message passing concurrency
- Use existing tools/libs
- Fault isolation
- Recoverable failure
- Hundreds of disk images

The Open Computer Forensics Architecture
(2001/2006-2012)

- Distributed
- Asynchronous
- Message passing concurrency
- Use existing tools/libs
- Fault isolation
- Recoverable failure
- Hundreds of disk images

- Poor page-cache usage → spurious reads
- CAS → spurious reads
- Spurious hashing → spurious reads.
- No locality of data design → spurious reads.

Challenges for message-passing concurrency based forensic processing

- Efficient use of available page-cache.
- Locality of data.
- Storage system & knowledge of active tool-chains.