

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 22-24 SEPTEMBER 2014

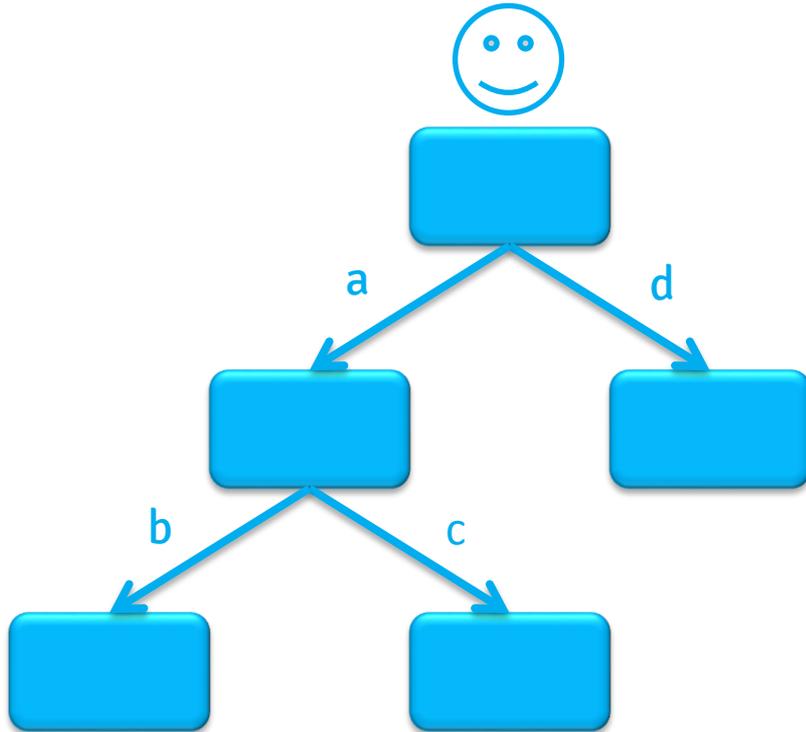
Oak, the Architecture of the new Repository
Michael Dürig, Adobe Research

Design goals

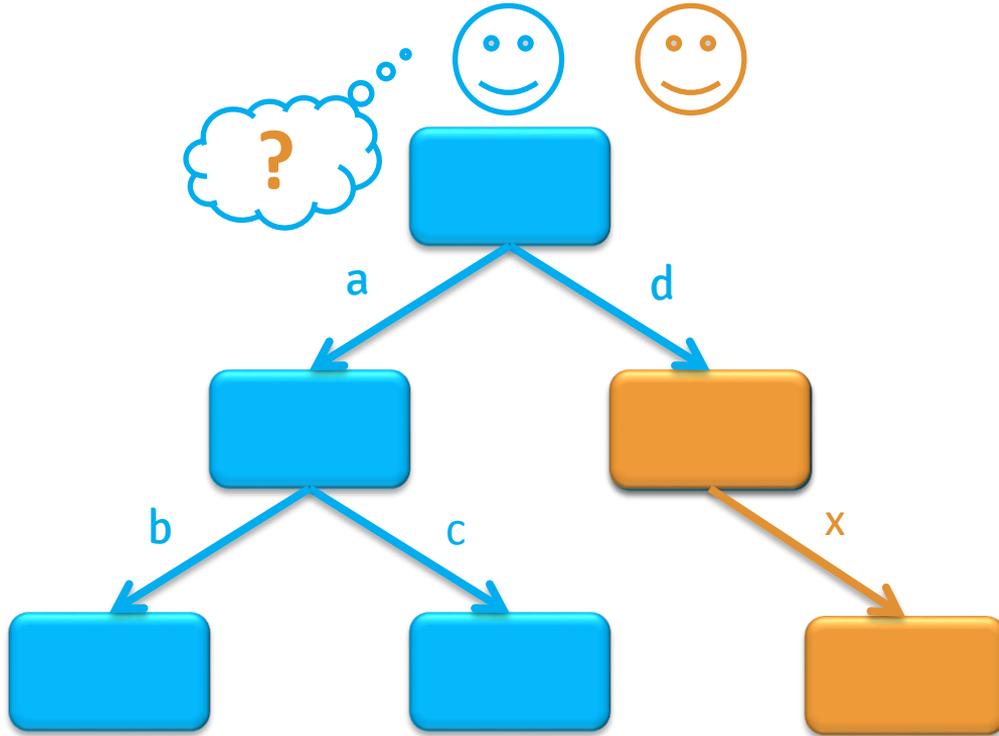
- Scalable
- Big repositories
- Clustering
- Customisable, flexible
- OSGi friendly

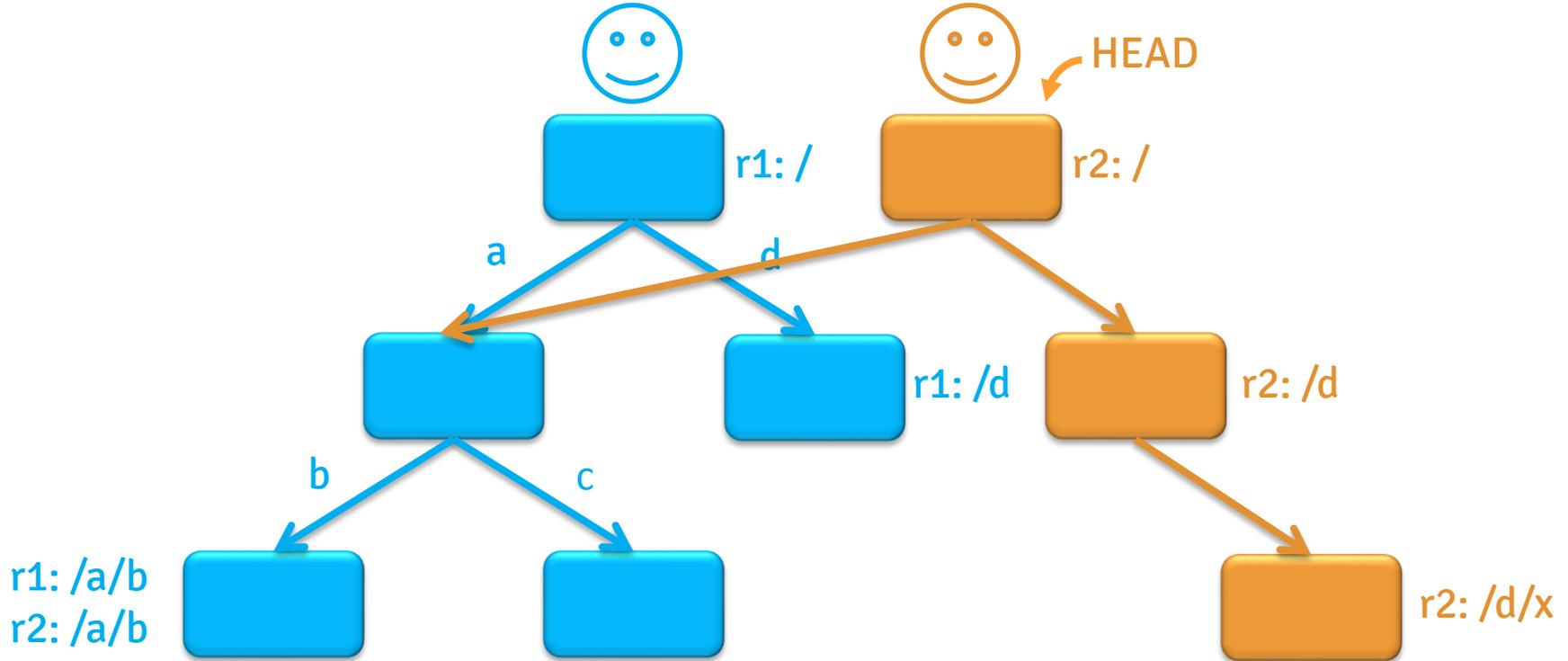
- CRUD
- Changes
- Search

Tree model



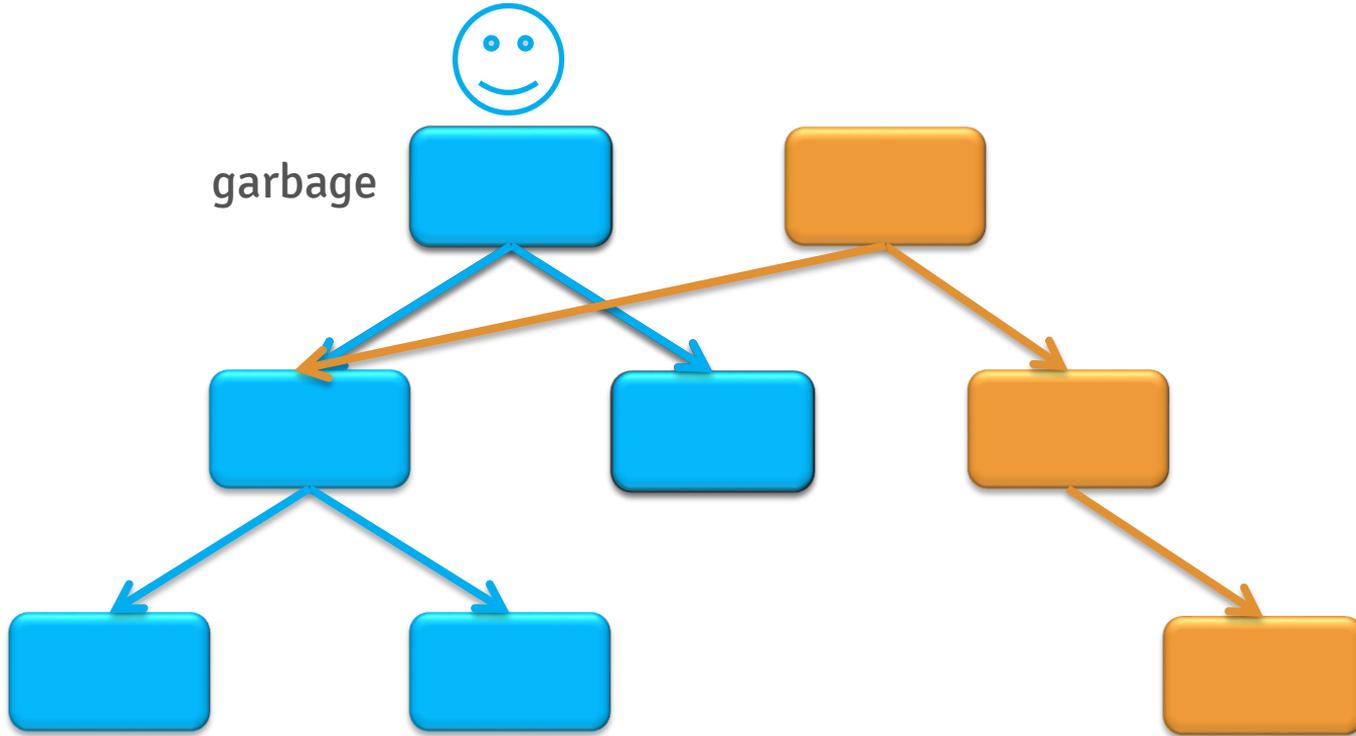
Updating



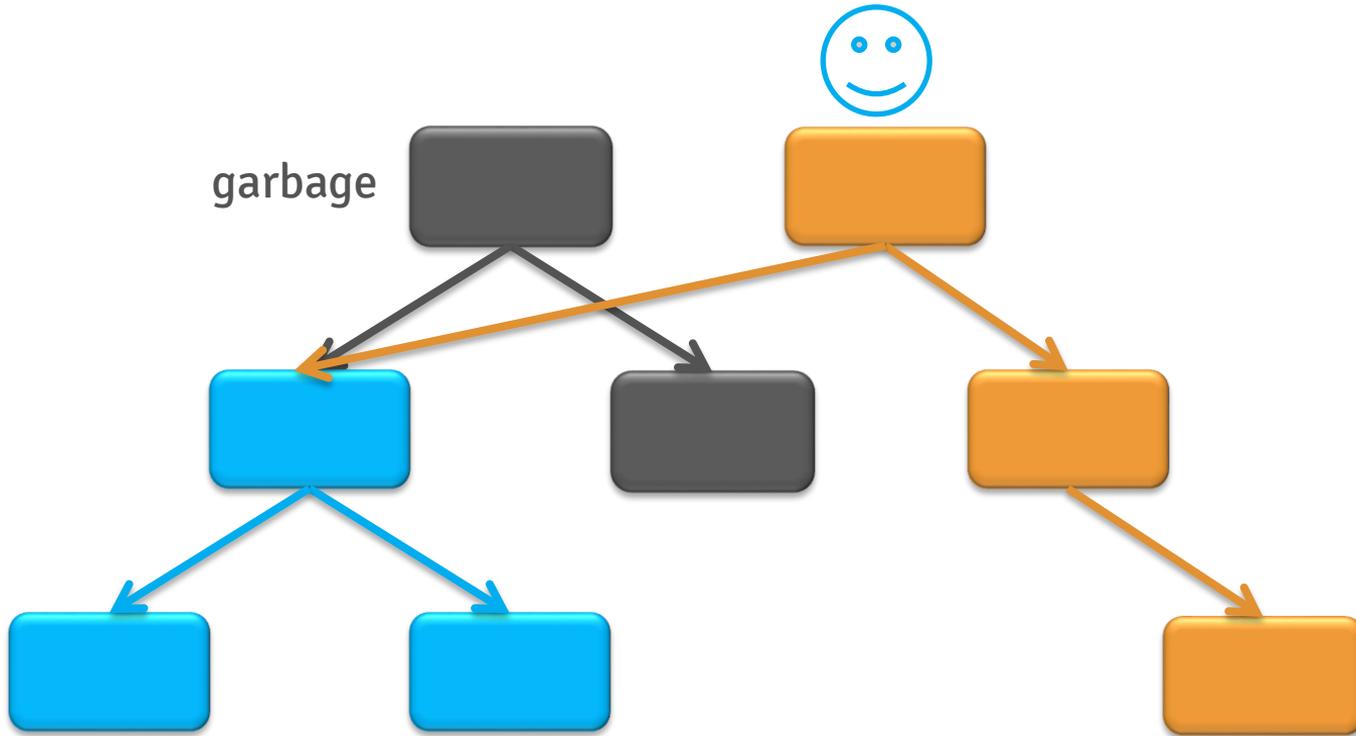


Refresh and Garbage Collection

Refresh

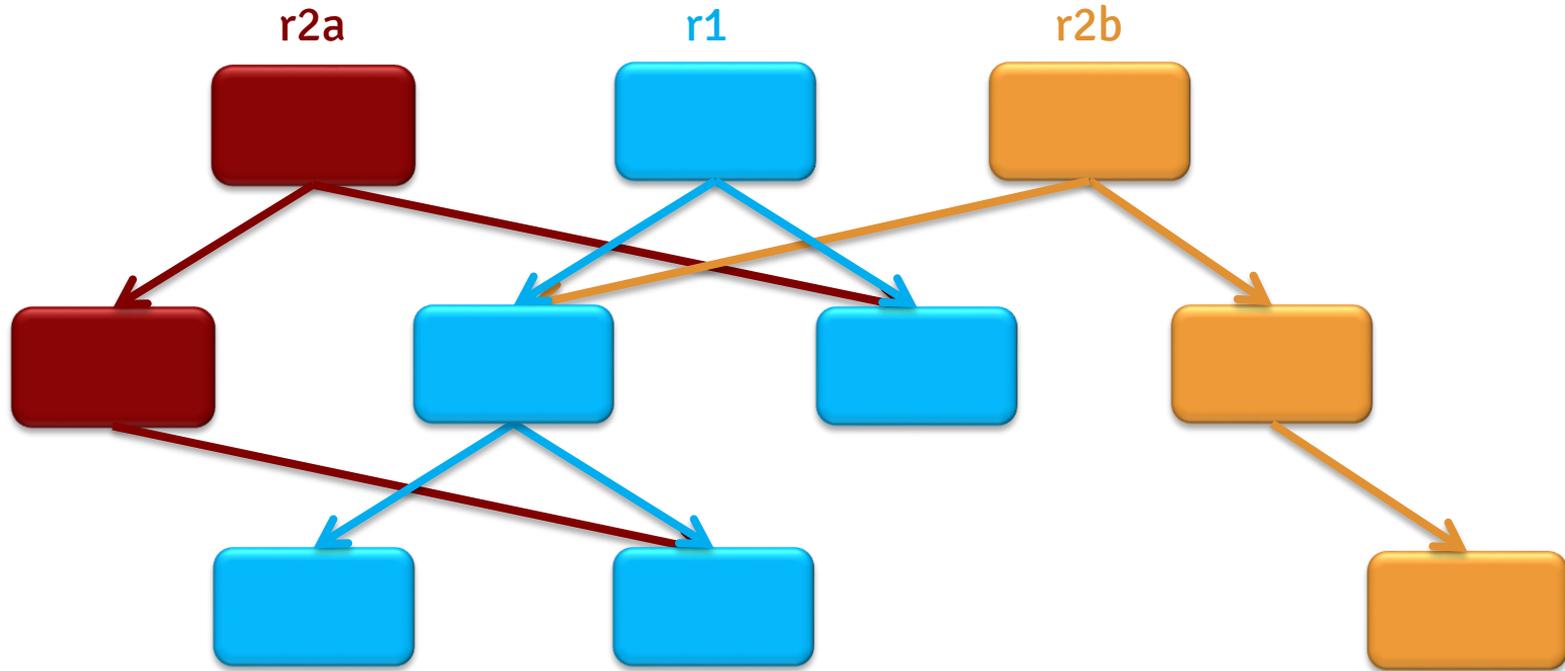


Garbage collection

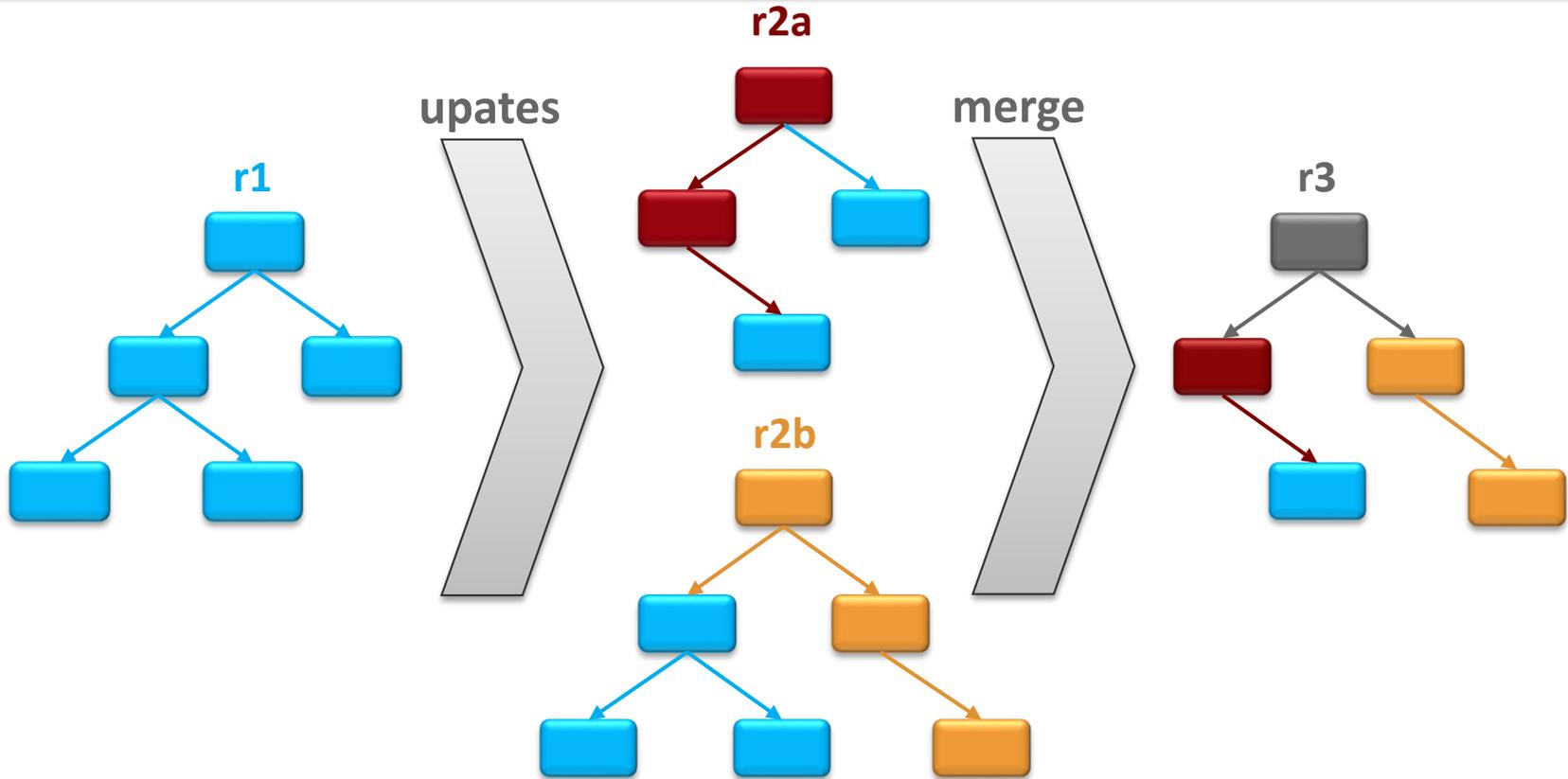


Concurrency and Conflicts

Concurrent updates



Merging



Conflict handling: serialisation

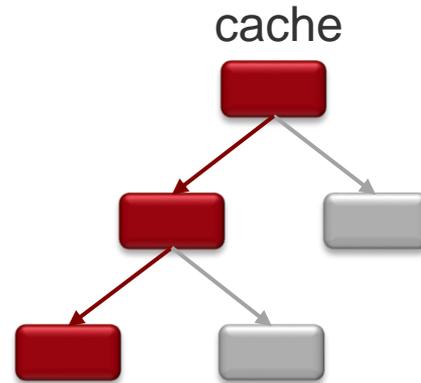
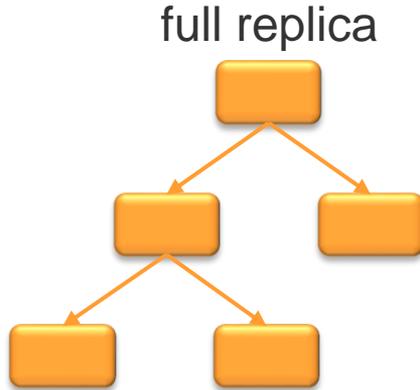
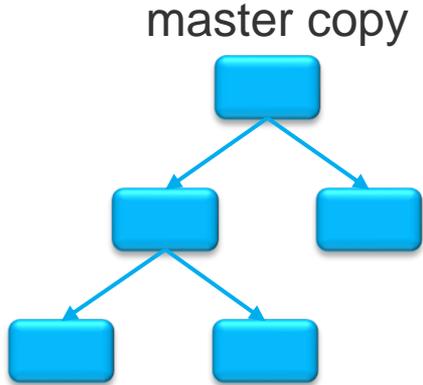
- **Fully** serialised
 - Fail, **no concurrent** update
- **Partially** serialised
 - Concurrent **conflict free** updates

Conflict handling strategies: merging

- **Partial** merge
 - Conflict **markers**, **deferred** resolution
- **Full** merge
 - Need to choose **victim**

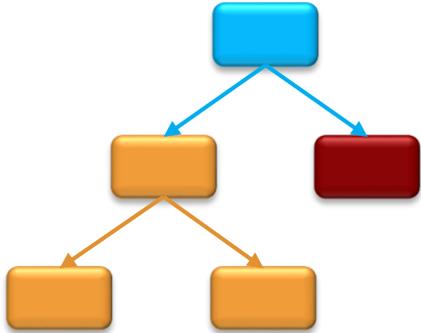
Replicas and Sharding

Replica and caches

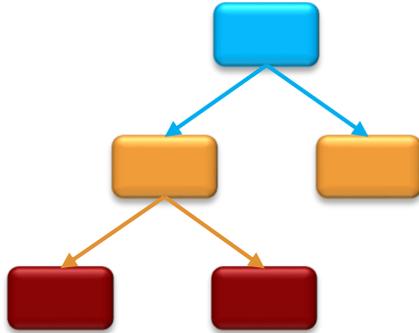


Sharding strategies

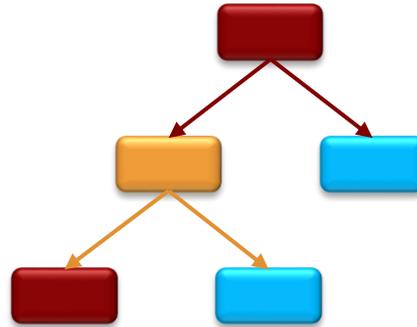
by path



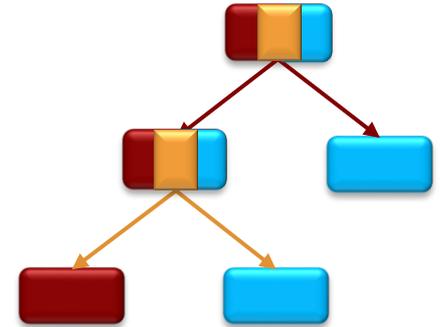
by level



by hash



with caching



Implementations

- Tree / Revision model implementation

Responsible for

Clustering

Sharding

Caching

Conflict handling

Not responsible for

Validation

Access control

Search

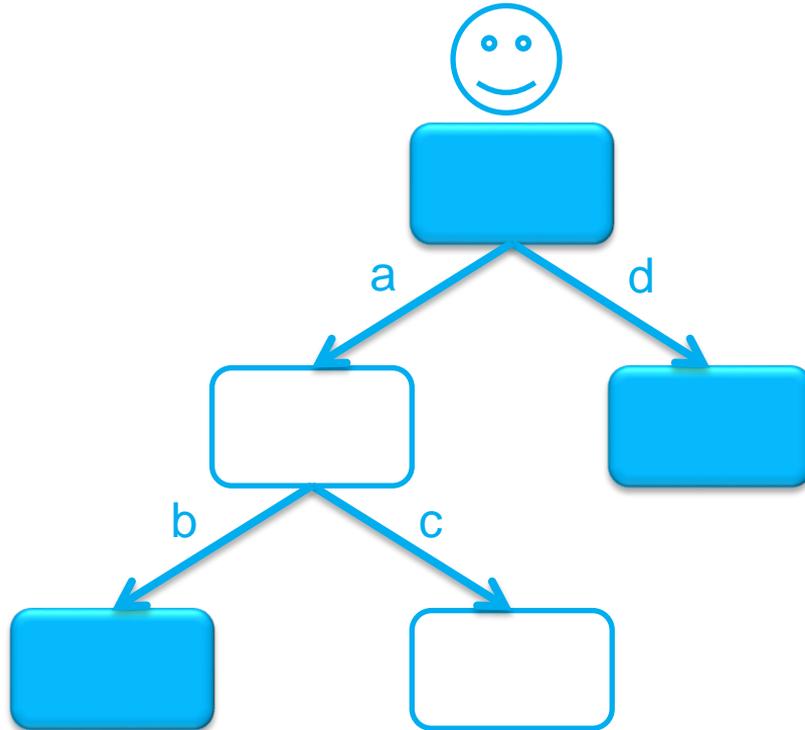
Versioning

Current implementations

	DocumentMK	TarMK (SegmentMK)
Persistence	MongoDB, JDBC	Local FS
Conflict handling	Partial serialisation	Full serialisation
Clustering	MongoDB clustering	Simple failover
Sharding	MongoDB sharding	N/A
Node Performance	Moderate	High
Key use cases	Large deployments (>1TB), concurrent writes	Small/medium deployments, mostly read

Access Control

Accessible paths



- All paths **traversable**
 - Node may not **exist**
 - **Decorator** on NodeStore

```
root.getChildNode("a").exists();
```

⇒ false



```
root.getChildNode("a")
```

```
  .getChildNode("b").exists();
```

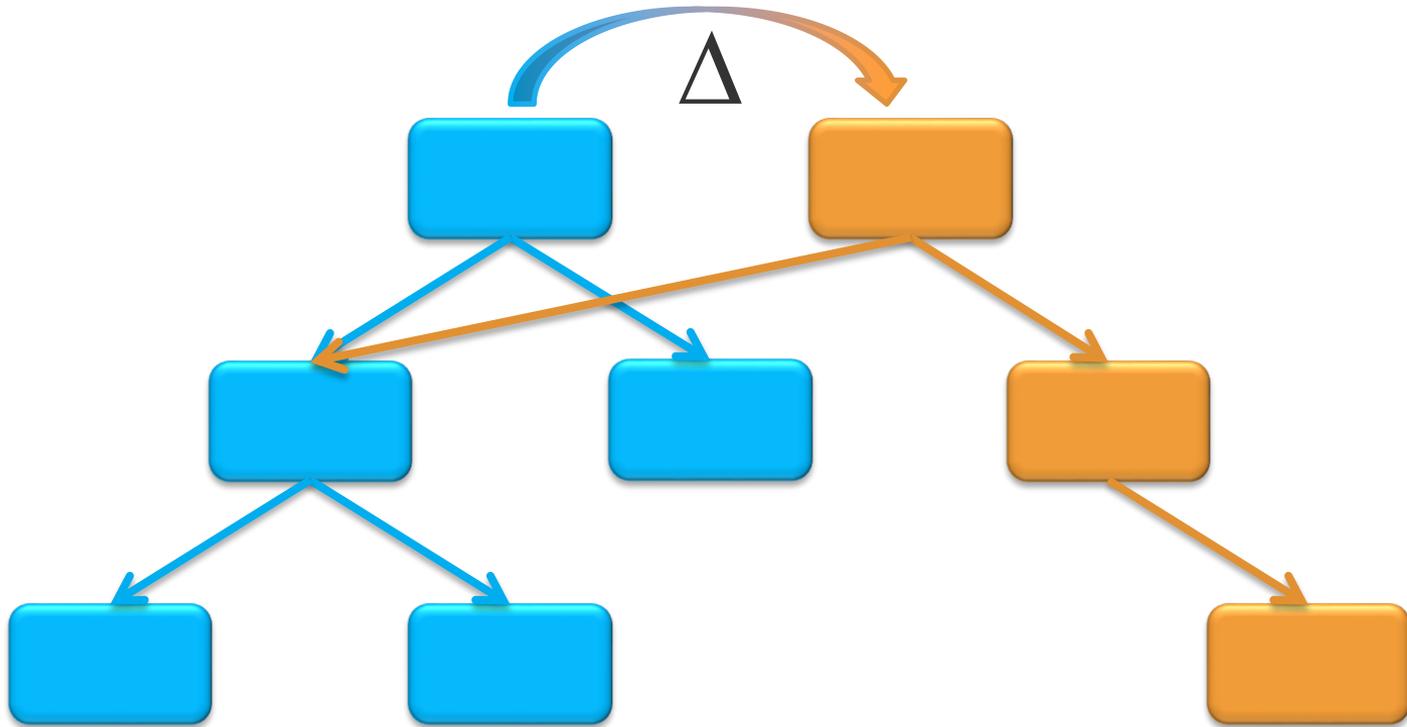
⇒ true



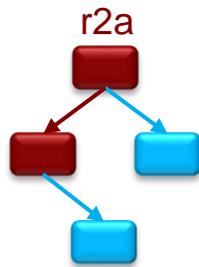
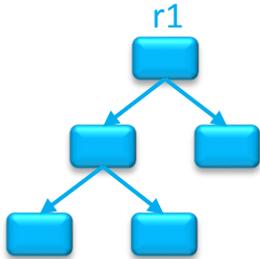
Comparing Revisions

- What **changed** between trees
- **Cornerstone** for
 - Validation
 - Indexing
 - Observation
 - ...

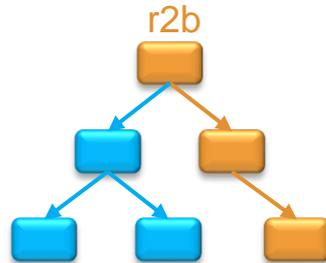
What changed?



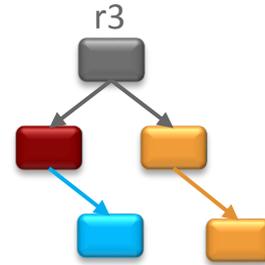
Example: merging



$r1 \xrightarrow{\Delta} r2a$
 “a” modified
 “b” removed



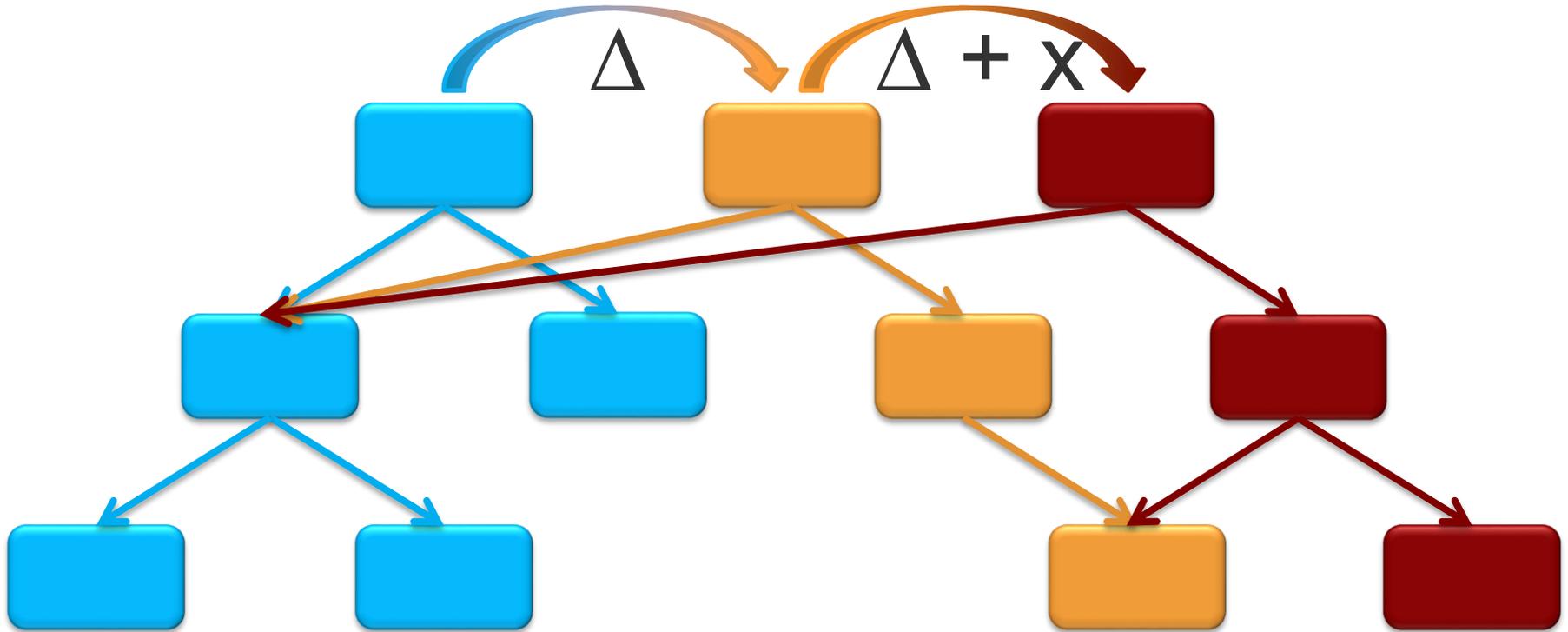
$r1 \xrightarrow{\Delta} r2b$
 “d” modified
 “x” added



Commit Hooks

- Key **plugin** mechanism
 - Higher level functionality
 - **Validation** (node type, access control, ...)
 - **Trigger** (auto create, defaults, ...)
 - **Updates** (index, ...)

Editing a commit



- Based on content **diff**
 - **pass** a commit
 - **fail** a commit
 - **edit** a commit
- Applied in **sequence**

Type of hooks

	CommitHook	Editor	Validator
Content diff	Optional	Always	Always
Can modify	Yes	Yes	No
Programming model	Simple	Callbacks	Callbacks
Performance impact	High	Medium	Low

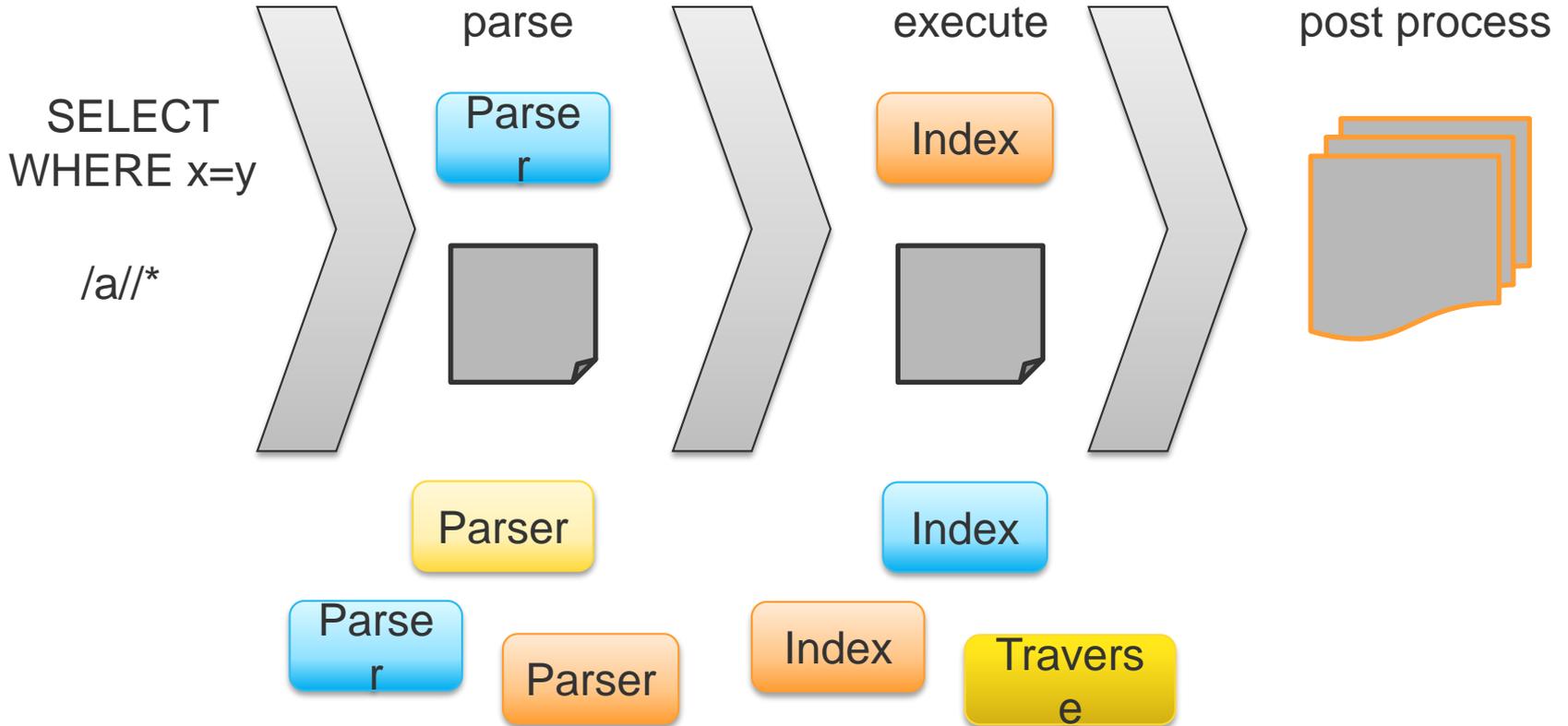
Observers

- Observe **changes**
 - **After** commit
 - Often does a content **diff**
 - **Asynchronous**
 - **Optionally synchronous**
 - Local cluster node only

- JCR **observation**
- External index **update**
- Cache **invalidation**
- Logging

Search

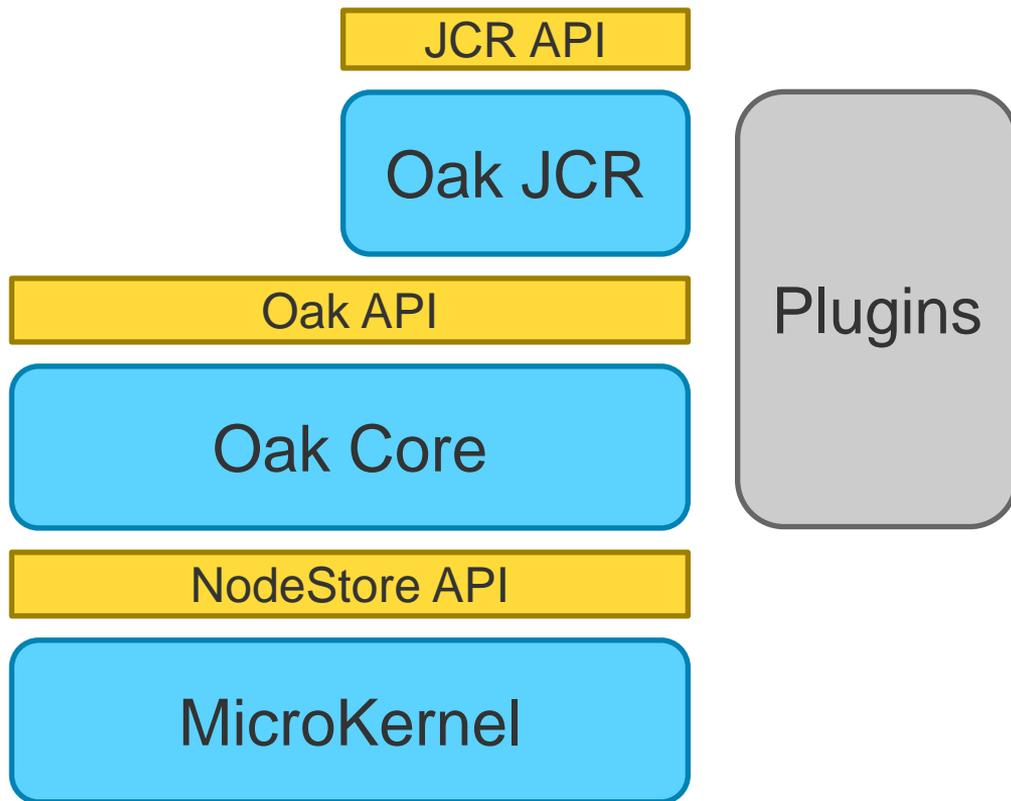
Query Engine



- Property (ordered)
- Reference
- Lucene
 - In-content or file system
- Solr
 - Embedded or external

Big Picture

Big picture



<http://jackrabbit.apache.org/oak/>



Appendix

<http://jackrabbit.apache.org/oak/>

<http://jackrabbit.apache.org/oak/docs/>

<https://svn.apache.org/repos/asf/jackrabbit/oak/trunk/>