



CaboLabs

Health Informatics, Standards and Interoperability

CaboLabs Platform



making clinical data work for you

Atomik and EHRServer

1. Both are openEHR Clinical Data Repositories (CDR)
2. Atomik will support demographics too (DDR)

Atomik and EHRServer

Both openEHR CDRs can:

1. receive data in openEHR formats (canonical JSON and XML)
2. validate data against openEHR schemas (syntactic)
3. validate data against openEHR templates (semantic)
4. manage data (versioning, audit logs, etc)
5. store and index data
6. query and retrieve data

Atomik and EHRServer

1. EHRServer is multi-tenant
2. Atomik is single tenant
3. EHRServer focus on SaaS (multi-client)
4. Atomik focus on single organizations

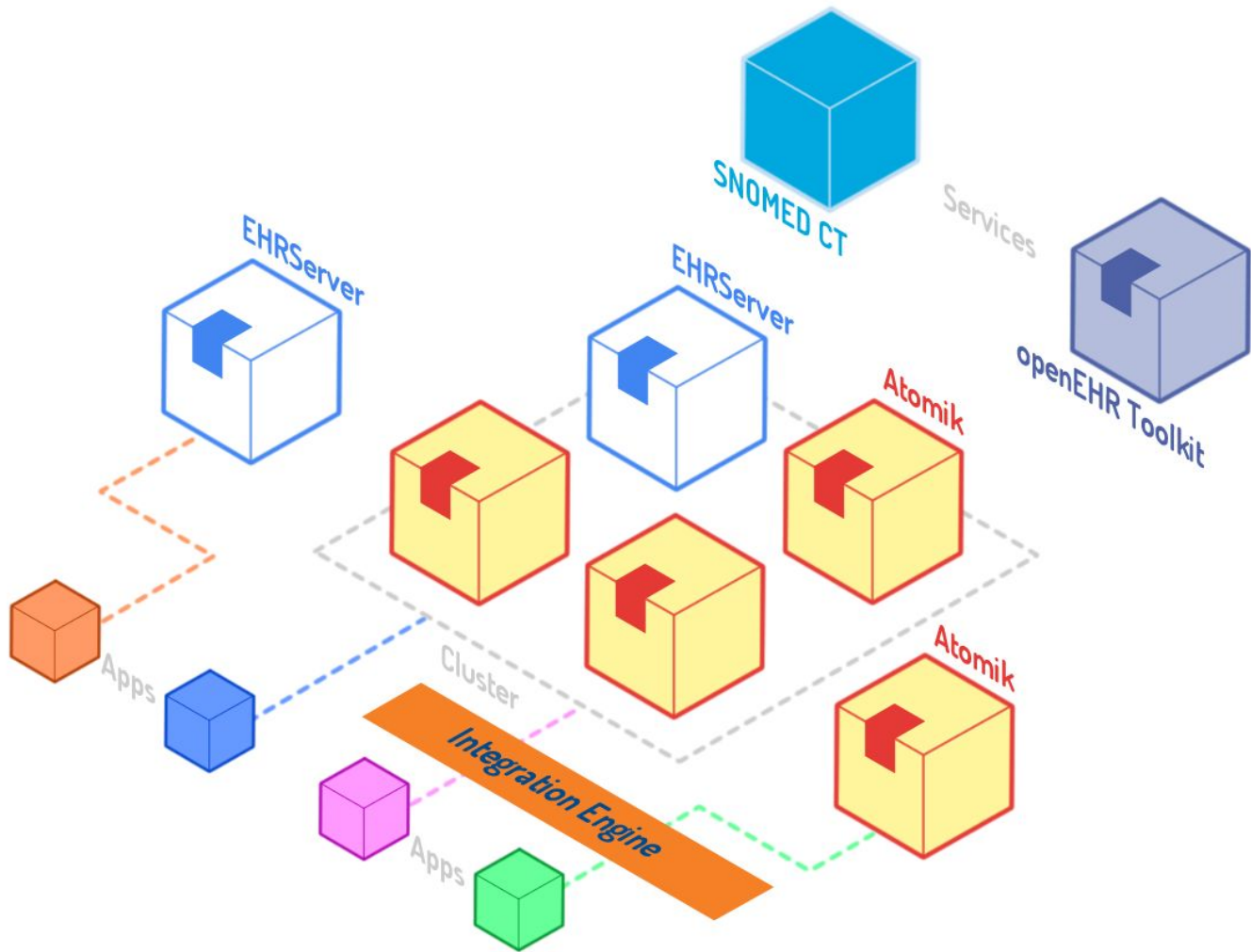
Atomik and EHRServer

1. Clustering supported at the system API level
2. One master/primary, many secondaries
3. Primaries accept writes, primary and secondaries accept query/retrieve
4. Use cases:
 - a. HA via load balancing of queries
 - b. automated backups
 - c. redundancy for fault tolerance
 - d. have separated query servers for research and education
 - e. ...
5. Sync inside a cluster has exact copies for all data
 - a. EHRs, COMPOSITIONs, Operational Templates, Queries, Users, etc.

CDR Configurations



flexible architecture



Atomik and EHRServer

Database

1. Don't use any feature of a specific database brand, just plain SQL
 - a. No specific functions, types or data structures
 - b. Doesn't use stored procedures
2. DB agnostic tech stack
 - a. Grails Framework implements data access via Hibernate
 - b. Really good abstraction layer
3. Can switch between any relational database that has a connector for Java
 - a. Adapts to what the client has and is used to
 - b. Don't impose the use of one type/brand of database

Atomik and EHRServer

Database

1. Most modern databases offer some kind of clustering/sharding mechanism
2. Allow horizontal scaling if data access is a bottleneck
3. Combine database clusters with Atomik/EHRServer clusters to support bigger loads, increase performance, and scale
4. Other techniques (out of scope) can be used like RAID at the physical level and preemptive caching at the app level

Atomik and EHRServer

Atomik is focused on high conformance levels with the openEHR specs

1. Though it doesn't implement AQL
2. Implements Simple Archetype Query Model (SAQM) querying formalism
 - a. Which is compatible with the openEHR REST API

I've been working in openEHR Conformance Verification for the last 4 years

1. Designed the openEHR Conformance Verification Framework
2. Test suites for technical verification of systems and test data sets
 - a. This is currently part of the openEHR Conformance Spec draft

Atomik and EHRServer

Why not AQL?

1. It's a general purpose query syntax that uses archetypes as an abstraction layer
2. The spec is incomplete, focuses on the syntax not on the specification of each operation, clause, function, etc
3. That generates different results for the same query on different implementations
4. Queries are not 100% portable
5. AQL imposes to use certain technologies and data structures, there is no freedom of implementation e.g. hierarchical structures
6. With and AQL implementation it's difficult to switch between different databases

Atomik and EHRServer

SAQM Queries

1. Supports SNOMED CT Expressions for semantic querying
2. Get datavalues or compositions
3. Combine single queries to filter/get EHRs
 - a. male patients
 - b. between 40 and 60 years old
 - c. with any type of diabetes (this uses SNOMED CT expressions to get all possible diabetes types)
 - d. with obesity

Other systems, tools and libraries in our platform

openEHR Toolkit: SaaS providing useful tools

1. archetype/template management
2. template dependencies (archetype roots tree)
3. template mindmap
4. data validation
5. data generation (most used tool, helpful for testing)
6. archetype flattening
7. ...
8. <https://toolkit.cabolabs.com/>

Other systems, tools and libraries in our platform

testEHR: for load testing

- https://www.cabolabs.com/blog/article/i_tried_to_kill_the_ehrserver-5a9a340db6ebf.html

controlEHR: dashboard and health monitor for multiple Atomik and EHRServer

openEHR SDK: reference model, OPT model, data validation, data generation, etc.

openEHR REST Client: Java/Groovy

EHRCommitter: sample client app to commit compositions

Other systems, tools and libraries in our platform

DICOM Broker: DICOM query/retrieve, WADO and SR creation (via HL7 v2.x)

openEHR Conformance Verification Framework

Education Program

- <https://www.cabolabs.com/education/>

Other systems, tools and libraries in our platform

PACS: we use DCM4CHEE

Integration Engine: we use Mirth Connect from NextGen

SNOMED CT services: we use Hermes

Identity Provider: we use KeyCloak

CaboLabs services

Architecture design

API design

Data model design

System/data integration

Coaching and support

Education

Consultancy