CaboLabs

What is EHRServer?

EHRServer is an open source, service-oriented, openEHR clinical data repository.

It provides a secure REST API to store and query clinical data in many ways, supporting standard formats like JSON and XML, that are easy to integrate with any front-end application.

Data queries can be created via the Administrative User Interface, with the powerful and easy to use EHRServer Query Builder.

EHRServer complies with the openEHR specifications[1], leveraging the openEHR Information Model and the Dual Modeling methodology, using standard Archetypes and Templates[2].

And it is open source, so you can customize it to your needs or you can collaborate helping with the development. It's license is Apache 2.

EHRServer was designed and developed after years of research and development of openEHR-based Clinical Information Systems, when we detected a niche for openEHR-compliant open source clinical data repositories. EHRServer was created by Pablo Pazos Gutiérrez[3] at CaboLabs Healthcare Informatics[4]

[1] http://openehr.org/releases/1.0.2/
[2] http://www.openehr.org/downloads/ADLworkbench/working_with_templates
[3] https://www.linkedin.com/in/pablopazosgutierrez
[4] http://cabolabs.com/en

Why EHRServer?

Unique
Currently there is no other system that for openEHR clinical data storage, that has a secure REST API and is open source.

Fast
Data commit and queries are executed in a few milliseconds.

Secure
The Administrative User Interface and the REST API can be easily secured by SSL Certificates, both requires authentication, and the REST API can only accept requests with self-signed tokens obtained after user authentication.

Generic
The EHRServer doesn't contain specific knowledge about the clinical records that will be stored.

Knowledge-driven
All the clinical records that will be stored in the EHRServer will be defined by standard openEHR Operational Templates, created from Archetypes.

Adaptable
The EHRServer can be adapted to different clinical contexts by configuring different sets of clinical document definitions (openEHR Operational Templates), and queries can be created from the Administrative User Interface.

Flexible
The EHRServer can be used on a wide range of contexts, from small clinics, to networks of hospitals, from hundreds of EHRs to tens of thousands.

Modifiable
To support new clinical documents and queries, no source code needs to be changed, nor the database schema needs to be changed. The EHRServer can be adapted to very different contexts, without changing the software.

Based on Standards
The main design concern of the EHRServer was to be compliant with the openEHR specifications, and use standard communication protocols and standard formats to move data in and out the EHRServer.

Interoperable
The use of standards, and a very well documented REST API, allows to integrate any application or system with the EHRServer in hours instead of weeks. OpenEHR Archetypes and Operational Templates guarantee Semantic Interoperability between the EHRServer and any system that makes use of it's data.

Accessible

Users and systems with permissions can access the clinical information contained in the EHRServer anytime, from anywhere. All the data that comes in can be queried, avoiding the "information silos", a very common problem in healthcare information systems. Also, if users don't have the information they need, how they need it, a specific query can be created and tested in seconds, using the EHRServer Query Builder, and new queries can be available in seconds for users.

Versionable

Because clinical documentation is inalterable, a versioning mechanism is needed to provider corrections or amendments to clinical documents. The EHRServer supports versioning of clinical documents and maintains all the versions of each document in a traceable structure.

Multitenancy

EHRServer supports to have different organizations, each EHR will be associated with one organization. This allows to support EHRs from many hospitals and clinics, on the same instance of the EHRServer. This is secure and very well delimited: one organization can't access the EHRs owned by other organization.

Intuitive Administrative User Interface (AUI)

The EHRServer AUI allows to manage, audit and track any aspect of the clinical records, EHRs, patients and queries. Allows to create patients and their EHRs and to create and test data queries. It also looks great on mobile devices!

Easy to setup and use

The EHRServer can be installed, configured and be running in less than 10 minutes. Read the EHRServer guide[5] to know more.

Made for the cloud

EHRServer can be easily deployed on the cloud on any PaaS provider that supports Java Web Applications, like OpenShift or AWS.

Well documented

The EHRServer Guide contains all the information you need to setup, run and use the AUI. It also contains the full REST API documentation.

Supported

The EHRServer is supported by CaboLabs Healthcare Informatics[6], experts on Healthcare Informatics, Interoperability and Standards, with many years of experience in R+D, consultancy and training.

[5] http://cabolabs.com/software_resources/EHRServer_v0.5.pdf

[6] http://www.cabolabs.com/en

What can you do with EHRServer? (main use cases)

EHRServer is designed to simplify the implementation of the following use cases, but is not limited in any way by them.

Shared Healthcare Record

It is very common that in a healthcare environment, like a clinic or hospital, multiple systems for clinical information recording are in use. It is also very frequent to have systems that are not designed to share information with other systems or clinical users, generating information accessibility problems because of the lack of interoperability, and fragmenting EHRs.

EHRServer can be used as an integration backend to share clinical information between those systems in an standard way, enabling data accessibility to clinical users and removing the EHR fragmentation.

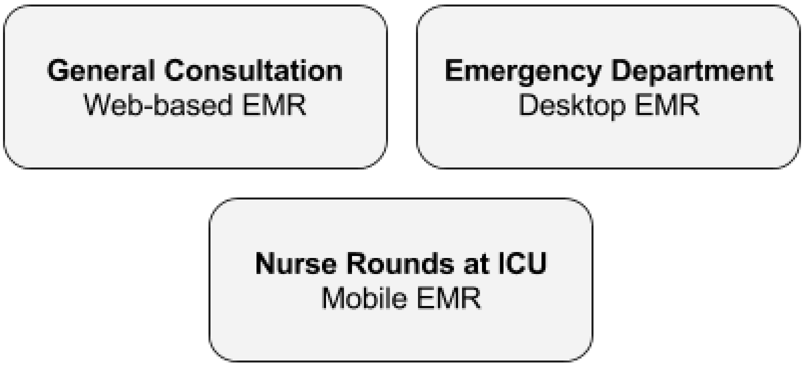
Typical scenario

A clinic or hospital has many systems to record clinical information in different ways, for different medical specialties or allied healthcare professionals, and different units or departments.

Those systems can be based on different technologies, platforms and devices, for example:

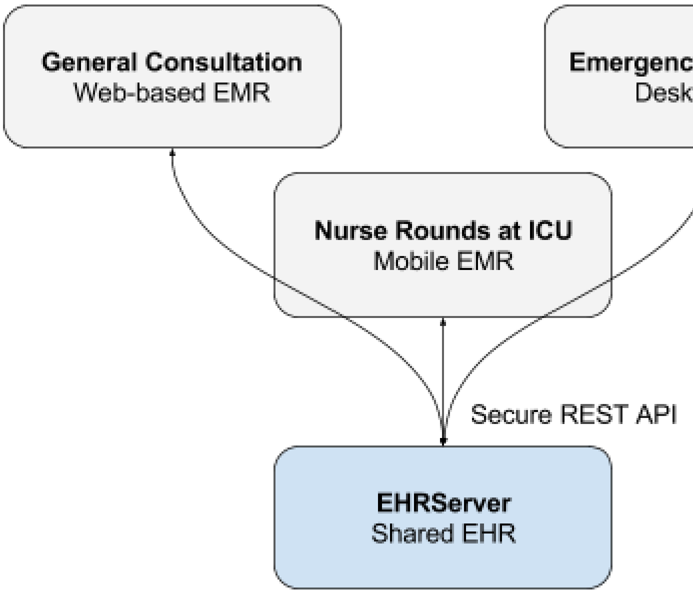
- a Web-based EMR for general consultation,
- a desktop EMR for the emergency department, and,
- a mobile application for the nurses to monitor patients at the ICU.

Clinical users will record information on those systems, but later they want a complete view over their patients, including the information from the three systems mentioned above, and other systems that might also be in place. For sake of simplicity let's keep just those three systems.



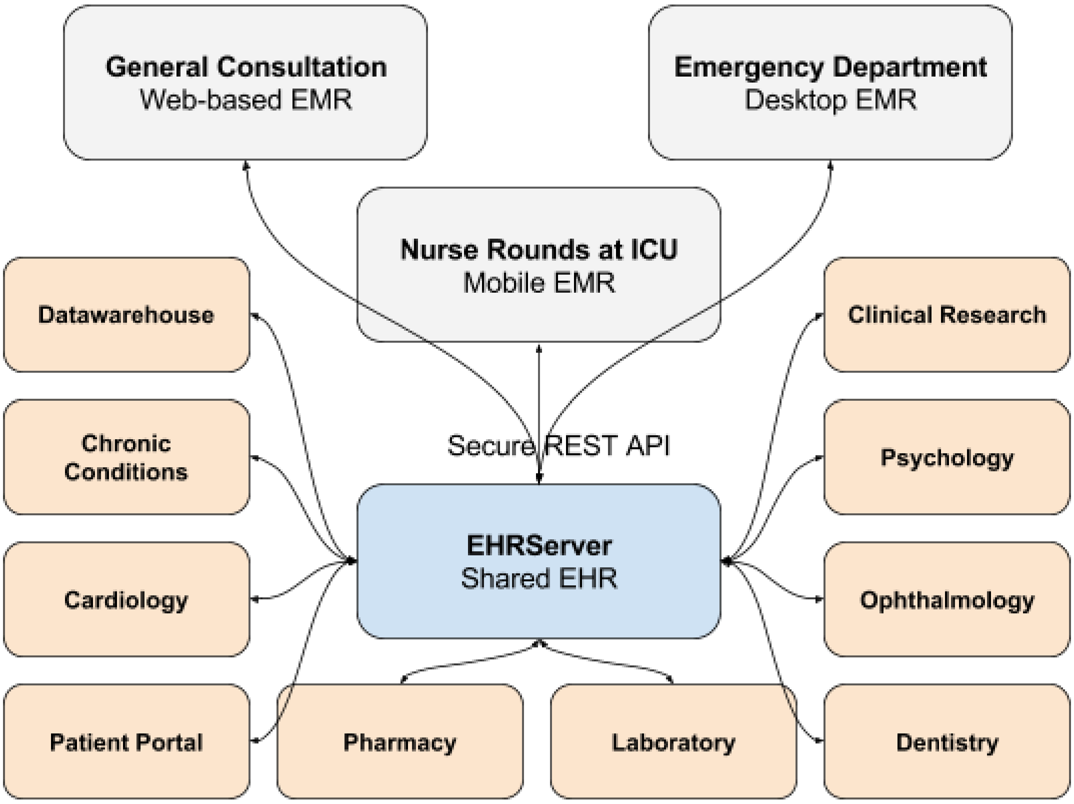
Level 1: no information sharing

First, let's share clinical information between those systems by integrating them with the EHRServer, using standard interfaces and data formats compliant with openEHR, instead of creating custom interfaces between each two of them (needs 2 interfaces to be implemented on each system).



Level 2: information sharing through EHRServer

Because there are no limits on what you can integrate, let's integrate more systems and apps that record clinical information, that display information, or for doing data analysis and research.



Level 3: sharing information with more applications, EHRServer as a platform

You will end up with an integrated and shared unique EHR per patient, accessible and interoperable.

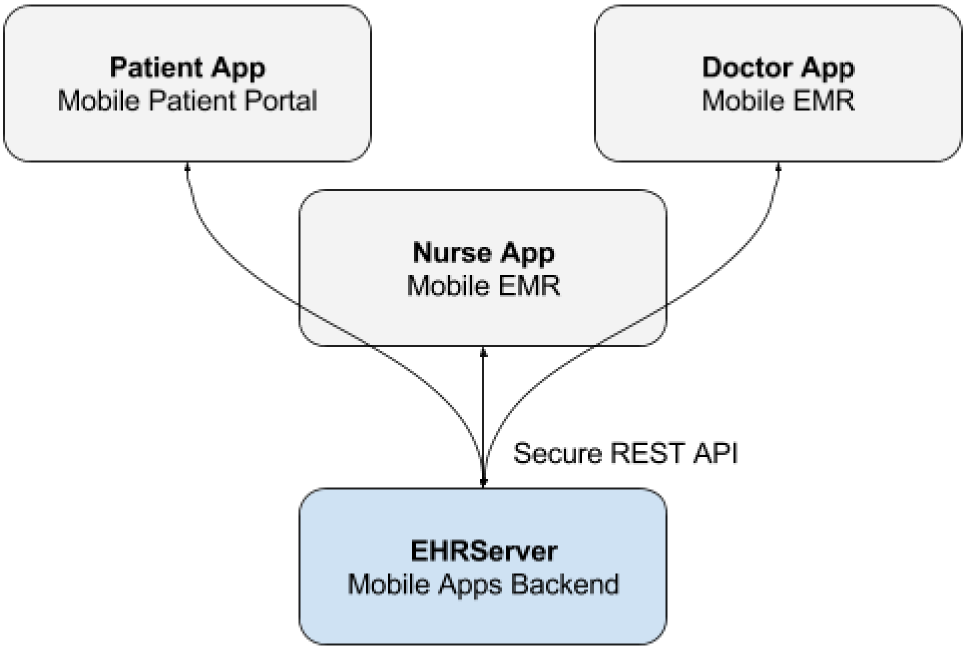
Backend for Clinical Mobile Applications

Many software factories that develop mobile applications don't have specific health IT knowledge in-house, so implementing an standard like openEHR is very difficult. Also implementing flexible clinical data storage is a hard task for non Health IT companies. EHRServer can alleviate the burden, all you need to know is how to use a REST API, XML and JSON, stuff that any software factory knows how to use.

With EHRServer as the backend of your healthcare mobile apps, you can focus on features and have you app running in no time, EHRServer takes care of the clinical data storage and the queries to access that data.

Multiple mobile apps can use the same backend, even if those apps are used by different hospitals and clinics, because EHRServer supports multi-tenancy:

- all clinical data is associated with an organization (clinic, hospital, etc.)
- users under each organization can access only data under their organization



EHRServer as backend of mobile apps

Clinical Research Datasource

Researchers usually have big sets of heterogeneous data to analyze, compare, chart, evaluate, etc. But there is no easy way to query the data sets in the many forms needed by researchers, and researchers end up using productivity tools like Excel to store and analyse the data sets, and do manual queries.

With EHRServer, the clinical data sets can be stored using a standard information model, that is easy to process and analyze, and the EHRServer Query Builder enable clinical researchers to create their own data queries to get the data they need, how they need it, allowing the finish the analysis in tools like Excel, where statistical analysis can be performed, but over a more specific data set.

Also, as shown on the previous use cases, EHRServer can be used to integrate many data sources to obtain more abundant data sets, without the burden of doing the data source integration manually (a complex process that requires a lot of time to get the desired results).

Distributed Clinical Data Store (under development)

In the near future, many instances of EHRServer will be able to be deployed forming a cluster. An EHRServer cluster will work as one logical EHRServer, composed by many physical EHRServers. This solution will enable these features:

- **High Availability**: 100% uptime of the EHRServer services, even if a server goes down.
- **Backup**: clinical data will be duplicated or triplicated between different servers, one can go down without any loss of data.
- **Disaster Recovery**: a current outage, fire, tornado, tsunami, etc. can affect the physical servers, but because of the redundancy of the cluster, no data is lost and the service is not interrupted. Later, new EHRServers can be added to the cluster and data loaded again to reach the same service level to users.
- **Scaling**: if more apps are added as clients of the EHRServer cluster, if those apps have more and more users every day, new EHRServer instances can be added to a cluster with ease to support scaling.

Main scenario for an EHRServer cluster

Two or more EHRServer instances can be added to a cluster. Each instance can receive data commits from client applications, and that data will be replicated to the whole cluster, so all the EHRServer instances will have the same data. That data will be available through queries over each EHRServer. So any client app can send and query information to / from each EHRServer in the cluster.

Also, other information like queries, users, organizations, ehrs, etc. will be synchronized inside each cluster.

Where can I find more info about EHRServer?

The EHRServer Guide
http://cabolabs.com/software_resources/EHRServer_v0.5.pdf

Demo of the latest version
https://www.youtube.com/watch?v=aHRDR5Vg2Hc&feature=youtu.be&t=2m10s

Source Code
https://github.com/ppazos/cabolabs-ehrserver

Questions?
http://www.cabolabs.com/forum/

If you need further clarifications:

Ing. Pablo Pazos Gutiérrez
pablo.pazos@cabolabs.com
www.CaboLabs.com