

Cloud Native Innovation Labs

AppMod Cloud Ready Demonstration

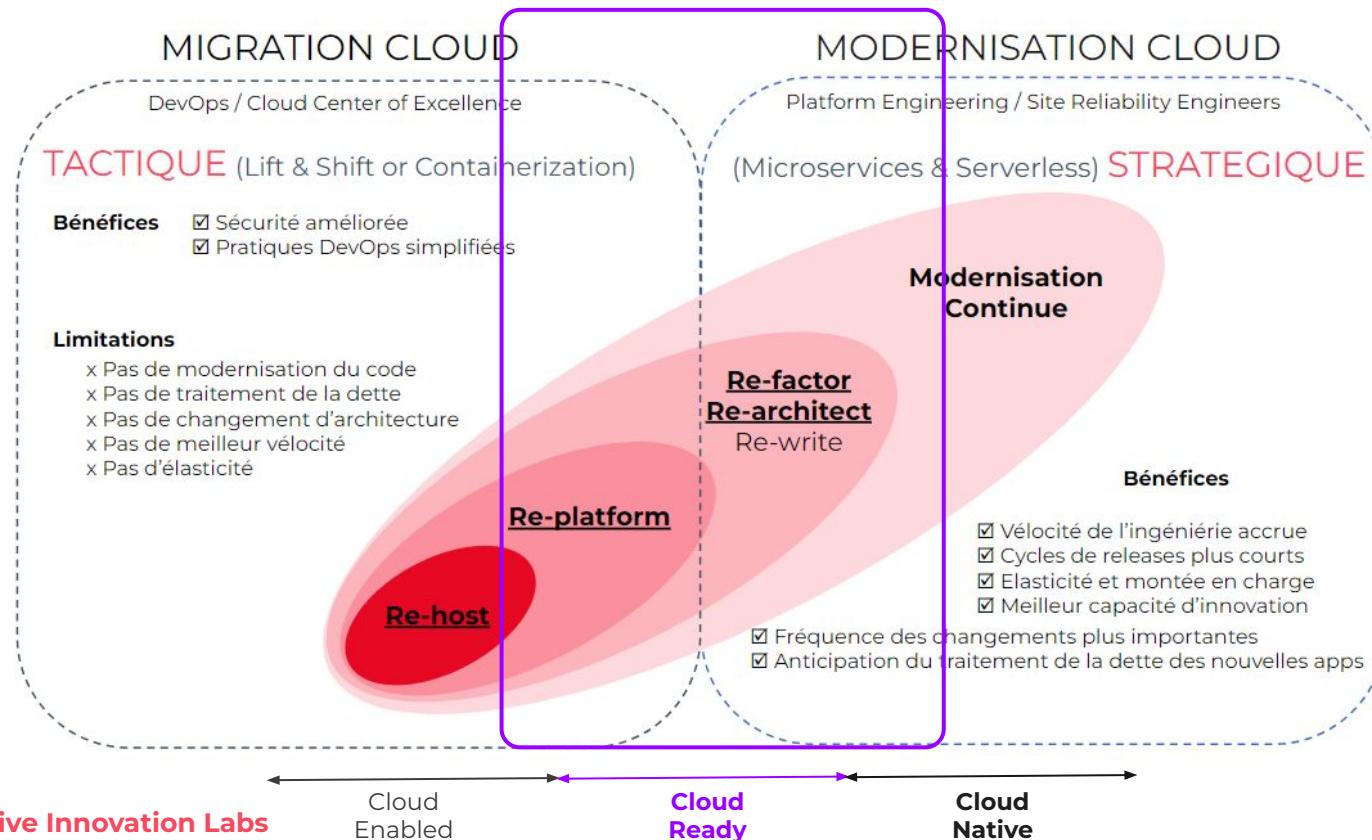
Introduction à la démonstration



Creative tech for Better Change

January 2024

Moderniser son patrimoine applicatif plutôt que le migrer



Les 5 types de modernisation possibles

1 REHOST - "Big Pod" Lift & Shift

Les applications patrimoniales sont déplacées telles quelles, sans réels avantages liés à l'utilisation du Cloud Natif. Les **conteneurs** sont considérés comme des machines virtuelles ou des "Big Pods".

Une attention particulière est accordée pour garantir une forte **cohérence des données** lorsque le "Big Pod" échoue. La plupart du temps, chaque Big Pod fonctionne en tant qu'instance unique.

Modernisation en CLOUD READY
Modernisation en CLOUD NATIVE

2 REHOST - "Light Pod" Lift & Shift

2 REHOST - "Light Pod" Lift & Shift

Les applications sont déplacées avec l'ambition de tirer parti des **services d'orchestration** Cloud Native de base tels que le dimensionnement automatique (vertical ou horizontal).

La migration est réalisée en exploitant des **facteurs de forme** de conteneur optimisés (légers, idéalement gérés via des opérateurs...).

3 REPLATFORM - Cloud Native Lift & Shift

Toutes les déploiements sont gérés en utilisant des pratiques Cloud Native telles que l'approche de déploiement **GitOps** ou les pratiques d'**observabilité**.

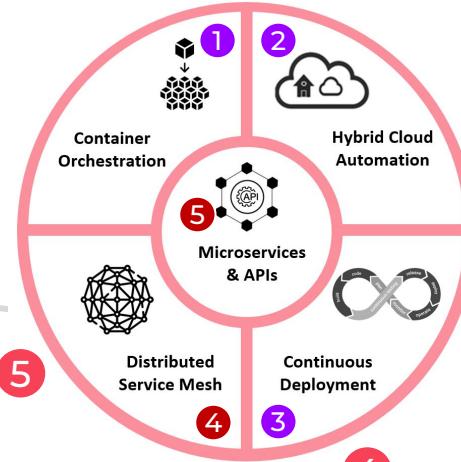
Des **services PaaS gérés** sont utilisés pour mutualiser les services backend tels que la base de données, le cache, la messagerie, etc.

4 REARCHITECT - Cloud Native Application

L'application est repensée et reconstruite en utilisant des **microservices** exposant des **API** via des protocoles communs tels que REST ou gRPC, nécessitant ainsi une couche de gestion des API.

Il est probable que la plupart des applications seront en **mode hybride** jusqu'à une refonte complète, sauf si elles sont reconstruites entièrement. Atteindre ce niveau permettra d'obtenir un **niveau élevé de résilience** (> 5 neuf).

Cloud Native Innovation Labs



4 REFACTORY - Cloud Native Refactoring

De nouvelles capacités sont mises en œuvre pour garantir une approche Zero Trust au sein du déploiement de l'application en utilisant un **ServiceMesh** tel que Istio.

Cela permet également des **approches de déploiement plus progressives** telles qu'un déploiement Blue/Green ou A/B, en supposant que la structure de l'application le permette.

L'application "Fil Rouge" : MedRec (J2EE)

The screenshot shows the MedRec application running in Mozilla Firefox. The title bar reads "MedRec Entrance - Mozilla Firefox". The address bar shows "localhost:7001/medrec/index.xhtml". The page has a header with links for "Weblogic", "Home", "Login", "Documentation", "Oracle", and "Console".

Administrator view and manage users.

The Administrator application allows an administrator to approve or deny new patient profile requests. The Administrator can reset passwords and create physicians.

[Login](#) [View Existing Users](#)

Patient look up and view your visit and prescription history and edit your profile information.

The Patient application allows patients to log in, edit their profile information, or request that their profile be added to the system. Patients can also view prior medical records of visits with their physician.

[Login](#) [I'm New Here](#)

1 2 MedRec Application (J2EE) 3

User Interaction: Web UI

Framework : J2EE 6

classes: ~ 120

LoC in classes: 13k

AppServer: Oracle WebLogic 12

Backend DB Derby

Hosting Docker Container

Modernization Type CLOUD READY

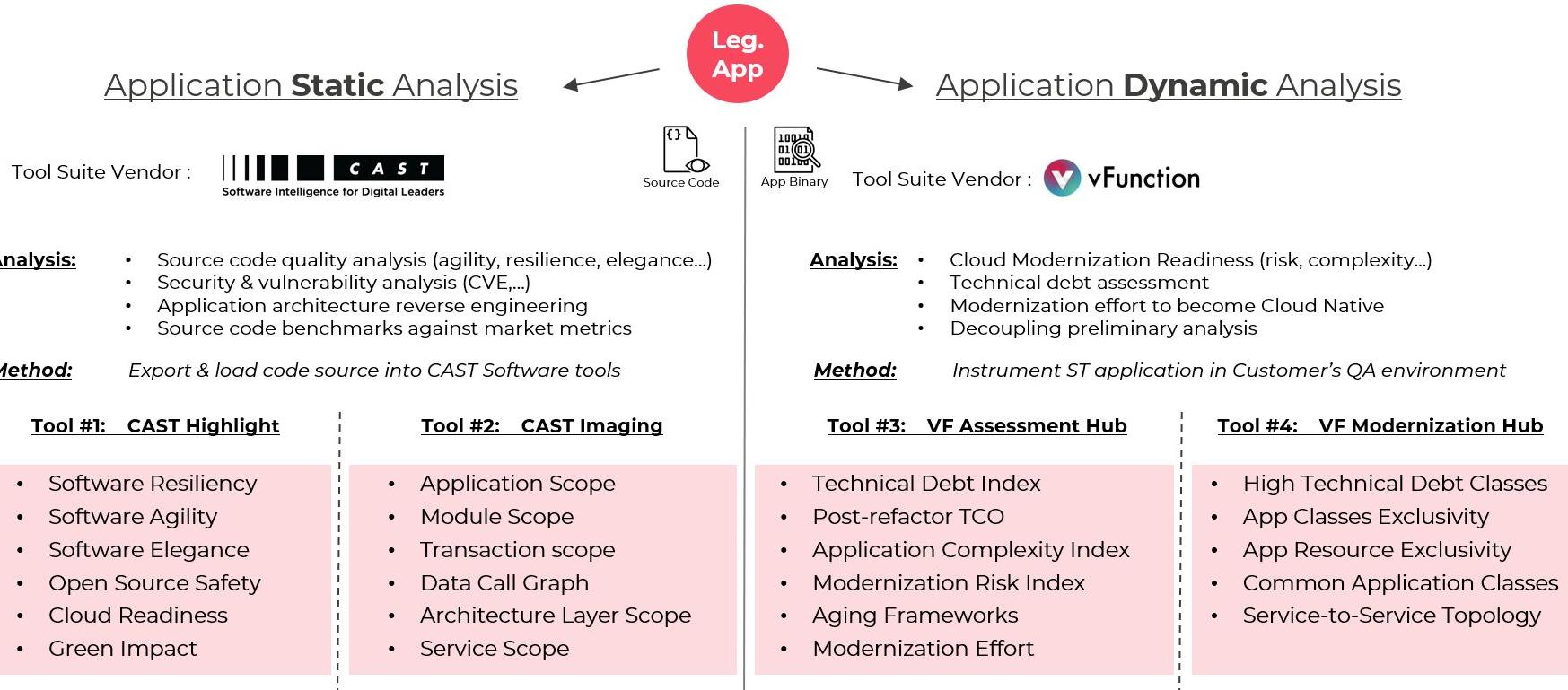
WAR deployments 3

Target Framework J2EE 6

Target db Backend Derby

Hosting Docker

Explication sur l'utilisation des outils de CAST et vFunction



Aperçu de la plate-forme vFunction

 **Architectural Observability Manager**

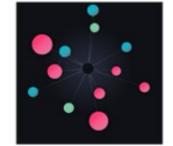
OBSERVE
AI Analysis & Baselining
(Detect Drift)

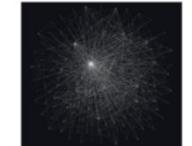




Config AI Parameters Domain Identification/Baselining Architectural Drift Thresholds

LEARN
Static & Dynamic Analysis
(Analyze Architecture)





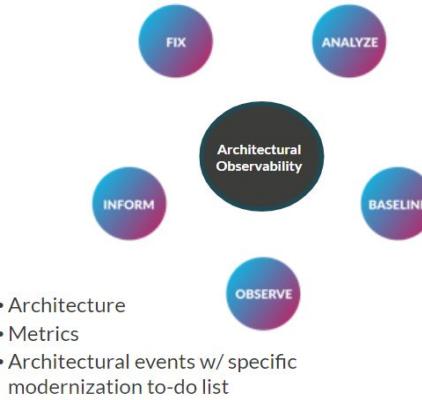
Resource Dependencies Domain's Call Tree Static Dependencies

ANALYZE
Static Analysis
e.g., CI/CD)





Tech Debt, Complexity, Risk Measure Tech Debt Impact Multi-App View



 **Refactoring Engine Add-on Module**

Code Refactoring



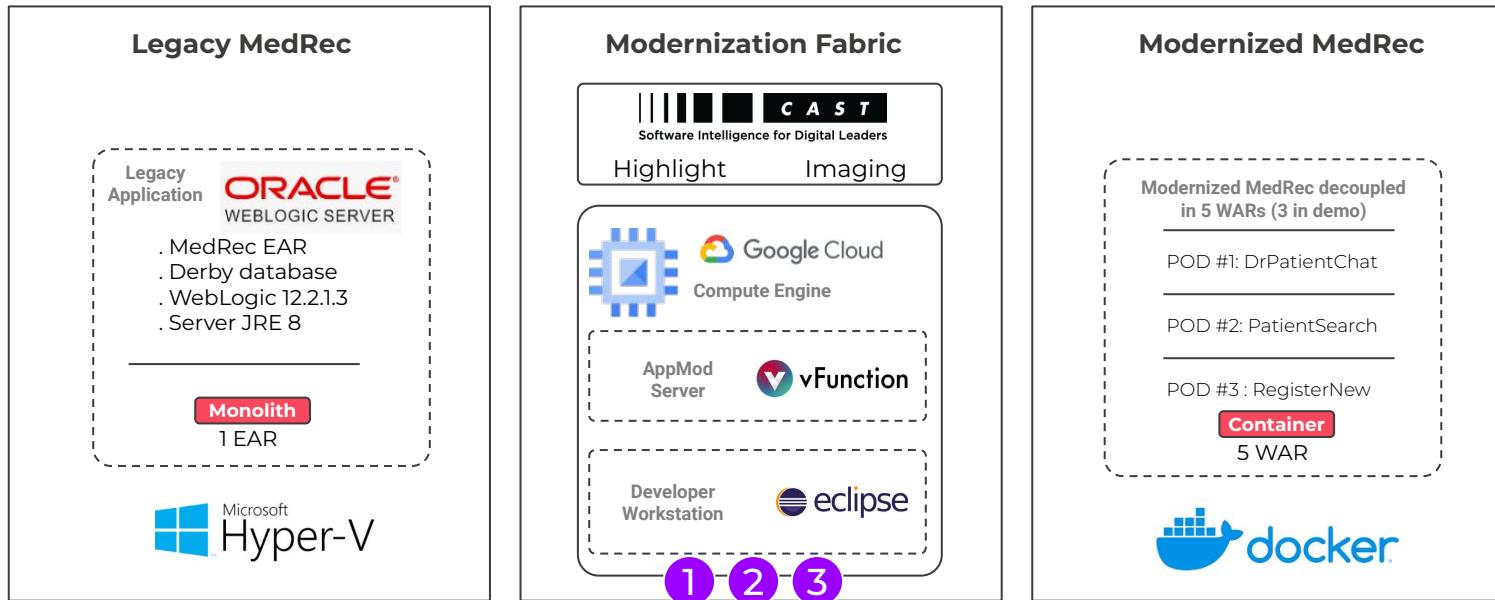
Service creation:
Service Decomposition



Service creation:

- Decomposition automation (microservices code)
- Frameworks upgrade
- Restful APIs generation
- Client libraries generation

Principes de modernisation de MedRec



Cloud Native Innovation Labs

AppMod Cloud Ready Demonstration

Scénarios de démonstration



Creative tech for Better Change





Software Health

Programming best practices that increase resiliency, improve agility and reduce complexity

71.8

WORST

66.3

INDUSTRY

63.37

BEST

77.2



Cloud Maturity

Software & organizational characteristics to optimize applications on the Cloud

64.1

WORST

59.7

INDUSTRY

62.68

BEST

68.4



Open Source Safety

Use of 3rd-party components that comply with security, license and age requirements

51.8

WORST

49.6

INDUSTRY

73.03

BEST

54.0



Green Impact

Programming practices and engineering principles that make software more environment-friendly

83.3

WORST

80.1

INDUSTRY

73.33

BEST

86.6

2/2

ONBOARDED APPLICATIONS

SEE PORTFOLIO DETAILS

0/0

CURRENTLY NO RUNNING CAMPAIGN

SEE PORTFOLIO TRENDS

11.0

TOTAL FTE

SEE DETAILS

17.2k

LINES OF CODE

SEE DETAILS

Quick Portfolio Insights

Top Applications to Invest Resources

Name	Segment
No data available in table	

No data available in table

See Software Maintenance

Top Applications to Refactor for Cloud

Name	Segment
MEDREC	REFACTOR

MEDREC

REFACTOR

See Cloud Maturity

Top Applications with urgent OSS Risks

Name	Segment
MEDREC	IMMEDIATE ATTENTION

MEDREC

IMMEDIATE ATTENTION

See Open Source Safety

Top Applications to improve Green Impact

Name	Segment
No data available in table	

No data available in table

See Green Impact

Top 20 Code Insights

MEDREC

MEDREC CodeScan_MEDREC

[Compare](#)[PPT Report](#)[Add to my Watchlist](#)[Overview](#)[Survey](#)[Health Distribution](#)[Code Insights](#)[Trends](#)[Cloud Maturity](#)[Green Impact](#)[Benchmark](#)[Software Composition](#)[Keywords](#) Rank 2 /2

Software Health

66.3

+ 0.0 %

 Rank 2 /2

Software Resiliency

75.3

+ 0.0 %

 Rank 2 /2

Software Agility

70.1

+ 0.0 %

 Rank 2 /2

Software Elegance

53.5

+ 0.0 %

 Rank 2 /2

Cloud Maturity

59.7

- 6.0 %

 Rank 1 /2

Green Impact

86.6

+ 0.0 %

 Rank 2 /2

Open Source Safety

49.6

+ 0.0 %

 Rank 1 /2

Component Security

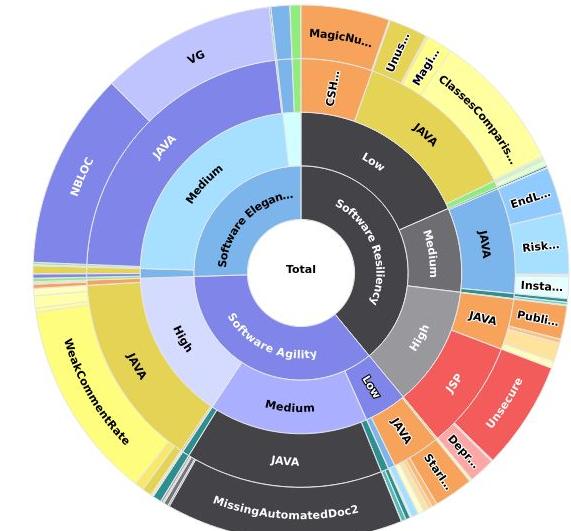
21.0

+ 0.0 %



Distribution

Portfolio Advisor for Technical Debt



Search:

Name	LOC	Files	BI	\$ Tech.Debt	\$ Tech.Debt/LOC	BFP	Total FTE	SR	SA	SE	Campaign Date
MEDREC	14.66k LOC	250	21	18.1 person-day	0.6 min/LOC	2k	8.00 FTE	75.29	70.07	53.53	01/30/24
OMS	2.56k LOC	89	34	4.2 person-day	0.8 min/LOC	48	3.00 FTE	85.60	79.89	66.19	01/14/24

Portfolio Advisor

Cloud Migration Waves

Health Factors

Cloud Requirements

Recommendations

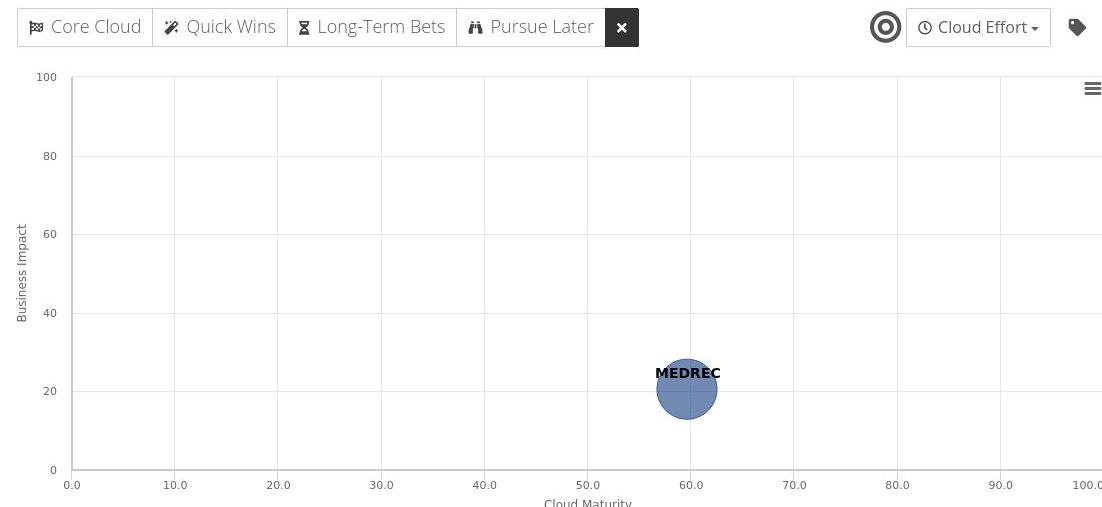
Containerization

Multi-Cloud

Portfolio Advisor for Cloud

Export

Dramatically speed up the time for developing a cloud migration portfolio roadmap with automatic recommendations on how to segment applications into different migration categories such as: Rehost, Refactor, Retire, etc. CAST Highlight includes the innovative Portfolio Advisor for Cloud, a dedicated portfolio-level report that automatically recommends migration segments for each application.



0
Rehost

A recommendation to change the infrastructure configuration of the application in order to "Lift & Shift" it to the cloud using Infrastructure as a Service

1
Refactor

A recommendation to perform modest modifications of the application code without changing the architecture or functionality so that it can be migrated to the cloud

0
Rearchitect

A recommendation to dramatically modify the application code thereby altering the architecture to improve the health of the application and enable it to be migrated to the cloud

0
Rebuild

A recommendation to discard the code of the application and develop it again in the cloud using Platform as a Service (PaaS) or serverless using Function as a Service

0
Retire

A recommendation to discard the application altogether or potentially replace it with a commercial Software as a Service (SaaS) alternative.

Portfolio Advisor

Cloud Migration Waves

Health Factors

Cloud Requirements

Recommendations

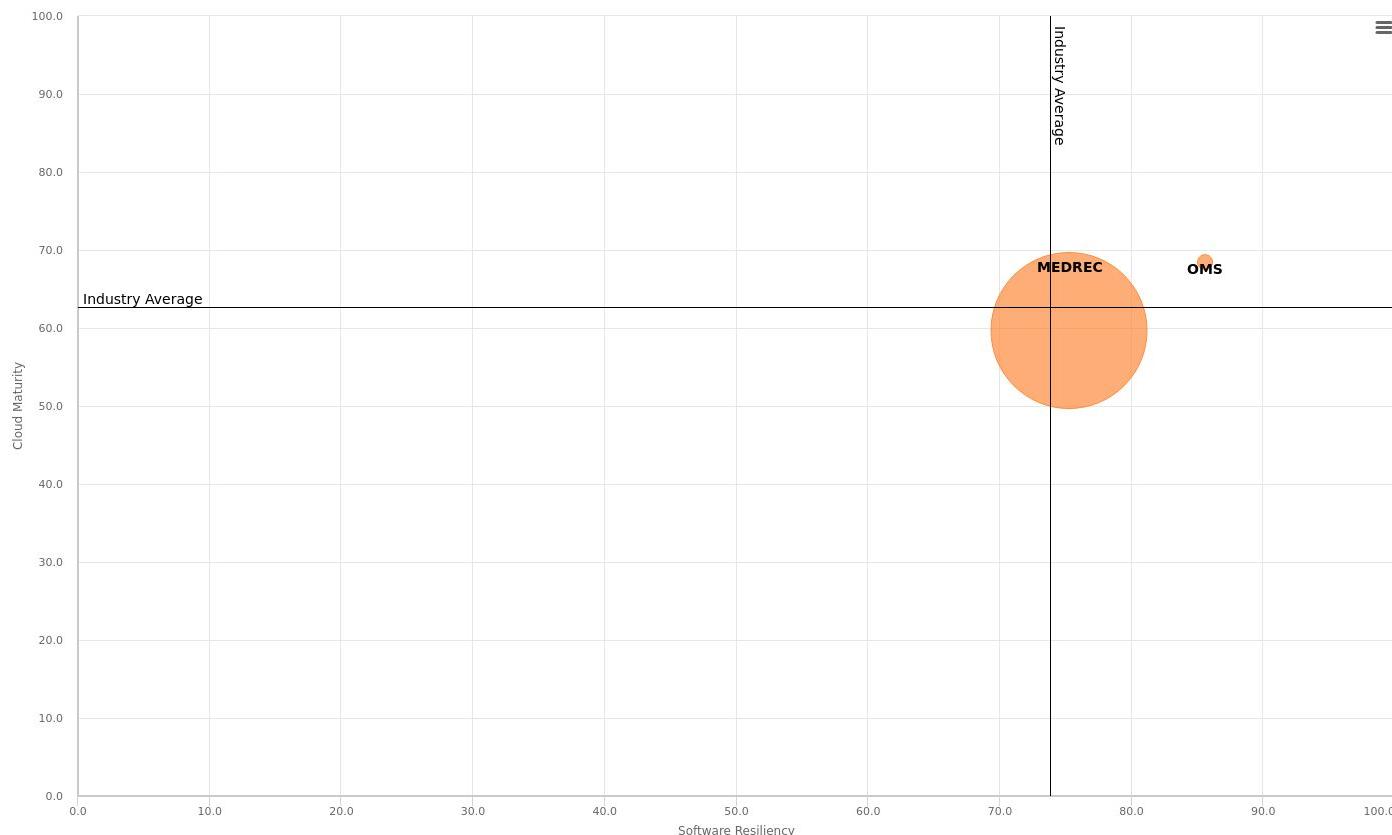
Containerization

Multi-Cloud



🛡 Software Resiliency 🚀 Software Agility 🎓 Software Elegance

🎯 Roadblocks ▾ 🔍 Pareto ⚙️



Portfolio Advisor

Cloud Migration Waves

Health Factors

Cloud Requirements

Recommendations

Containerization

Multi-Cloud

Cloud Boosters

Cloud Requirement

No data available in table

Technology

Total apps

Cloud Blockers

Cloud Requirement

Technology

Impact

Criticality

Roadblocks

Est. Effort ?

Total apps

Security & User Authentication : Use of an unsecured data string ⓘ

Python

Critical

2

0.25 person-day

1

Security & User Authentication : Use of an unsecured data string ⓘ

Java

Critical

1

0.13 person-day

1

Persistent Files : Using stateful session (Servlet) ⓘ

Java

High

5

0.10 person-day

1

VMWare Tanzu Execution Environment : Using Oracle WebLogic ⓘ

Java

High

129

16.13 person-day

1

VMWare Tanzu Execution Environment : Using JEE Configuration ⓘ

Java

High

1

0.13 person-day

1

Persistent Files : Perform Directory Manipulation ⓘ

Java

Medium

1

0.03 person-day

1

Execution Environment : Using file system ⓘ

Java

Medium

18

0.56 person-day

1

Persistent Files : Perform File Manipulation ⓘ

Java

Medium

5

0.16 person-day

1

Execution Environment : Using file system ⓘ

C#

Medium

18

0.56 person-day

1

Execution Environment : Using file system ⓘ

Python

Medium

1

0.03 person-day

1

Security & User Authentication : Hardcoded URLs using HTTP protocol ⓘ

Java

Low

1

0.03 person-day

1

Portfolio Advisor

Cloud Migration Waves

Health Factors

Cloud Requirements

Recommendations

Containerization

Multi-Cloud

Export to Excel

✓ Cloud Service Recommendations ?

Using Software Intelligence, CAST Highlight combines detected code patterns, technologies and OSS components and contextual characteristics of an application to help cloud architects identify the best cloud native services to adopt for their custom applications from Cloud Service Providers.

Platform : Google Cloud (3) ▾

③ Eligible Google Cloud Cloud services

✓ Cloud Storage

Object storage for companies of all sizes. Store any amount of data. Retrieve it as often as you'd like.

1 Applications

MEDREC

Technical Documentation

Get started with Cloud Storage

✓ Cloud Run

Develop and deploy highly scalable containerized applications on a fully managed serverless platform.

2 Applications

MEDREC OMS

Portfolio Advisor

Cloud Migration Waves

Health Factors

Cloud Requirements

Recommendations

Containerization

Multi-Cloud

Containerization Insights

The table below lists Cloud Maturity patterns that have been triggered and could block or slowdown containerization of applications

Cloud Requirement	Technology	Impact	Criticality	Roadblocks	Est. Effort	Total apps
Persistent Files : Using stateful session (Servlet) 	Java	C A	High	5	0.10 person-day	1
Persistent Files : Perform Directory Manipulation 	Java	A	Medium	1	0.03 person-day	1
Execution Environment : Using file system 	Java	A	Medium	18	0.56 person-day	1
Persistent Files : Perform File Manipulation 	Java	A	Medium	5	0.16 person-day	1
Execution Environment : Using file system 	C#	A	Medium	18	0.56 person-day	1
Execution Environment : Using file system 	Python	A	Medium	1	0.03 person-day	1
Security & User Authentication : Use of unsecured network protocols (HTTP, FTP) 	C#	C	Low	1	0.02 person-day	1
Security & User Authentication : Use of unsecured network protocols (HTTP, FTP) 	Java	C	Low	1	0.02 person-day	1

[Execution Environment](#) [Agnostic](#)

Using file system

Cloud applications should not assume the local file system is accessible, as the directory structure might be different from a traditional desktop or server machine and/or the Cloud application may not have sufficient rights to access the local file system. Instead, use relative paths to application resources (e.g. `../../reporting/reportBuilder.xml`). Depending on your application context and the Cloud platform where it is deployed, you could also consider using functions or classes like `LocalResources` to dynamically resolve file paths.

Criticality

[Blocker](#) [Medium](#)

The criticality level impacts the application Cloud Maturity score, whether it is a blocker (a negative pattern found during the code scan) or a booster (pattern that makes your app more Cloud-ready) and the corresponding level (low, medium, high, critical).

Migration Impact

[ARCHITECTURE](#) [CODE](#) [FRAMEWORK](#)

Migration impact levels indicate the different layers of your application that you should modify or fix, in order to comply with Cloud-oriented programming best practices and remove the identified roadblocks.

Containerization

[Criticality](#) [Medium](#)[Migration Impact](#) [ARCHITECTURE](#)

Container applications should not assume the local file system is accessible, as the directory structure might be different from a traditional desktop or server machine and/or the application may not have sufficient rights to access the local file system. Instead, use relative paths to application resources (e.g. `../../reporting/reportBuilder.xml`). Use the CAST Highlight output listing all your code files declaring dependencies to the filesystem. Review the dependencies in each file. Also assess if this file system dependency for execution will be needed by multiple containers. If it is the case, consider the use of volumes instead of writable layers. Note: When many roadblocks are identified for this blocker, they may be addressed holistically with one remediation.

References

<https://docs.docker.com/storage/> <https://www.baeldung.com/ops/docker-container-filesystem> <https://medium.com/@BeNitinAgarwal/docker-containers-filesystem-demystified-b6ed8112a04a>

Searched Code Pattern

[Agnostic](#) [C#](#)

Look in source code strings that contain OS-specific paths:

`C:\, D:\ ... \\Server\` for Windows platforms
`/bin, /boot, /dev, /etc, /home, /lib, /lib64, /mnt, /opt, /proc, /root, /run, /bin, /srv, /sys, /tmp, /usr, /var` for Linux platforms

[Agnostic](#) [C/C++](#)

Look in source code strings that contain OS-specific paths:

`C:\, D:\ ... \\Server\` for Windows platforms
`/bin, /boot, /dev, /etc, /home, /lib, /lib64, /mnt, /opt, /proc, /root, /run, /bin, /srv, /sys, /tmp, /usr, /var` for Linux platforms

[Agnostic](#) [COBOL](#)

Look in source code strings that contain OS-specific paths:

`C:\, D:\ ... \\Server\` for Windows platforms
`/bin, /boot, /dev, /etc, /home, /lib, /lib64, /mnt, /opt, /proc, /root, /run, /bin, /srv, /sys, /tmp, /usr, /var` for Linux platforms

[Agnostic](#) [Clojure](#)

Flux de démonstration

1. Step 1: Présenter les 2 cas d'usage MedRec et ce qu'on va obtenir à la fin (préciser new UI à développer)
2. Step 2: Présenter l'interface graphique de vFunction en utilisant leur instance de démo car + riche
3. Step 3: Basculer sur l'instance locale de vFunction et présenter MedRec avant discovery + static analysis
4. Step 4: Lancer une découverte et réaliser les 2 cas d'usage présentés + chat
5. Step 5: Charger une découverte complète, basculer sur Analysis et présenter les bubbles + focus sur le PatientChatWebSocket (essaye d'expliquer « au mieux » + préciser que ce n'est pas magique !)
6. Step 6: Charger le SNAP-002 et préciser que l'objectif est de passer de la vue éclatée à la vue consolidée (préciser les compétences nécessaires + le jus de cerveau nécessaire) + extraction des codes sources des 3 webapps avec la target JEE 7
7. Step 7: On bascule sur le studio Eclipse + explication de comment c'est obtenu + montrer rapidement les fichiers modifiés pour le service PatientSearch
8. Step 8: On termine en présentant la création des containers et en montrant le test des APIs REST
9. Step 9: OpenMind, expliquer comment cela aide à simplifier la modernisation, les nouvelles features, le cycle de release, les opérations ...

Cloud Native Innotation Labs

Step 1

Présenter les 2 cas d'usage MedRec et
ce qu'on va obtenir à la fin (préciser
new UI à développer)



devoteam

Welcome to Avitek Medical Records World

Avitek Medical Records (or MedRec) is a WebLogic Server sample application suite that demonstrates all aspects of the Java Platform, Enterprise Edition (Java EE). MedRec is designed as an educational tool for all levels of Java EE developers. It showcases the use of each Java EE component, and illustrates best-practice design patterns for component interaction and client development. MedRec also illustrates best practices for developing and deploying applications with WebLogic Server.

[Getting Started!](#)

The MedRec User Type

To begin using MedRec, click the **Start Using MedRec** button below. From there, you can begin by registering as a new patient, or you can log in using one of the previously defined roles listed below.

The administration username and password are the ones you entered when creating your Weblogic Server samples domains. Use this username when logging into the WebLogic Server Administration Console.

- **Patient** - The Patient application allows patients to log in, edit their profile information, or request that their profile be added to the system. Patients can also view prior medical records of visits with their physician.
- **Admin** - The Admin application allows an administrator to approve or deny new patient profile requests.
- **Physician** - The Physician application allows physicians and nurses to log in, search and access patient profiles, create and review patient medical records, and prescribe medicine to patients.

[Start using Medrec »](#)

Additional Resources

Use these resources to learn more about the Avitek Medical Records sample application:

- [WebLogic Server News and Community](#)

You can post and read questions, comments, or suggestions about MedRec in

Documentation

Avitek Medical Records (or MedRec) is a comprehensive sample application that demonstrates how to use Oracle WebLogic Server and Java EE features together in a real-world Java EE application suite. In addition to providing extensive code examples for developing Java EE applications, Avitek Medical Records helps you to understand the Oracle WebLogic Server development environment and how to use that environment to build your own applications and modules. If you are already familiar with Oracle WebLogic Server or basic Java EE programming, or if you are interested in using Oracle WebLogic Server development tools, start with the Avitek Medical Records application and documentation.

[View details »](#)

Database Connectivity

Data for the MedRec applications is stored in a Derby database installed with WebLogic Server. You can view the Derby data directly by running the Derby SQL scripting tool, ij, located in the WL_HOME/common/derby/bin subdirectory of the WebLogic Server installation directory. The URL of the MedRec database is



Administrator

view and manage users.

The Administrator application allows an administrator to approve or deny new patient profile requests. The Administrator can reset passwords and create physicians.

[Login](#)[View Existing Users](#)

Patient

look up and view your visit and prescription history and edit your profile information.

The Patient application allows patients to log in, edit their profile information, or request that their profile be added to the system. Patients can also view prior medical records of visits with their physician.

[Login](#)[I'm New Here](#)

Physician

look up a patient, add visit information, and prescribe medications.

The Physician application allows physicians and nurses to log in, search and access patient profiles, create and review patient medical records, and prescribe medicine to patients.

[Go To The Physician App](#)

Cloud Native Innotation Labs

Step 2

Présenter l'interface graphique de
vFunction en utilisant leur instance
de démo car + riche



4 Total Apps | Jan 22, 2024

Default View ▾

Sort By: App Name ▾

Debt Level (1 - 100)

1 100

Accumulation of Debt:

- 0 +

App Size:

XS S M L XL

App Type:

Java .Net

Last Updated: ▾

Created: ▾

Has Analysis

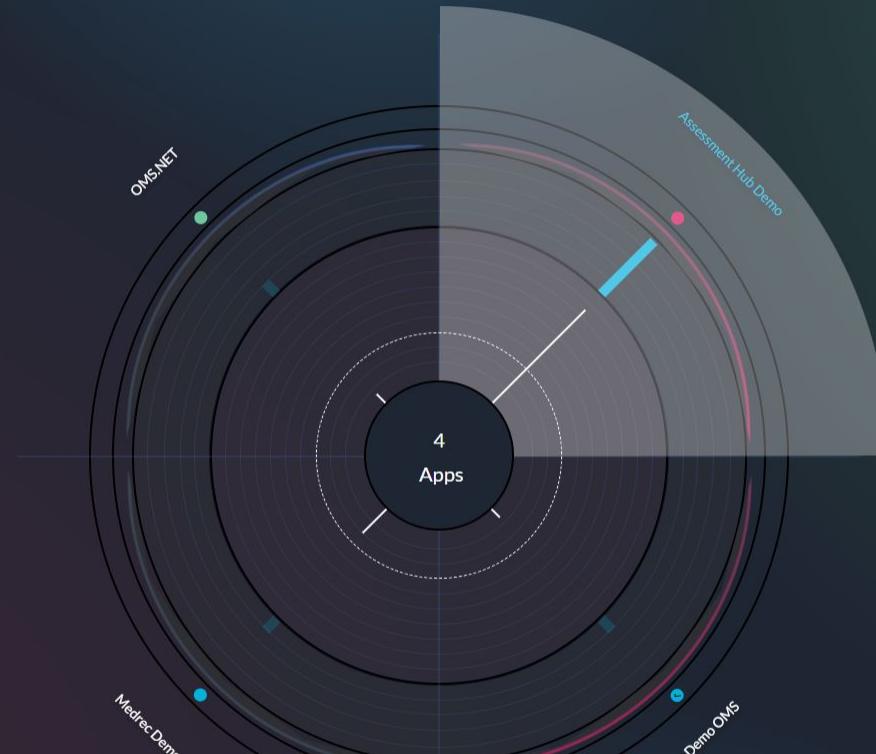
Architectural Complexity

1 2 3 4 5

✓ (Denotes Modernized)

Application Tags ▾

Download Filtered Report



Assessment Hub Demo

App Last updated: 09/01/2024

View Tech-Debt Report

Debt	(Avg. All)
86.54%	32%
Accumulation of Debt	
Neutral	

Size of App
18359 classes
XLarge

Aging Framework
82%

Architectural Complexity
Very high

Application Tags (0) ⚡

TODOs	
0	
Manual TODOs	0 (0 Done)
Drift TODOs	0 (0 Done)

Hide Legend ▾

Debt Size (1-100)

Color denotes accumulation

- 0 +

App Size

XS, S, M, L, XL

4 Total Apps | Jan 22, 2024

Onboarding Status

INSTALLATION	LEARNING	ANALYSIS	OBSERVATION									
0 Apps	1 App	2 Apps	1 App	4 Schedule Runs	0 Dynamic Class Exclusivity Alerts	1 Domain Changes Alert	0 Dead Code Alerts	0 High Debt Alerts	0 Common Library Alerts	1 Resource Exclusivity Alert	1 Architecture Complexity Alert	
0 Static Data Available	0 Actively Learning	0 Active Analysis	1 Missed Scope	0 Static Class Exclusivity Alerts	0 Library Circular Dependencies							
0 Controllers Installed	1 Analysis Ready	2 Baseline Ready	1 Missed Trigger									
0 Learning Ready												

SERVICE CREATION

3 Apps

8 Services Created

Licensed Classes	18,612
Learning Time	47h
Analysis Events	7,421



Onboarding

INSTALLATION	LEARNING	ANALYSIS	OBSERVATION	SERVICE CREATION
0 Apps	1 App	2 Apps	1 App in Observation Phase	3 Apps

Favorites Installation Learning Analysis Observation Search Architecture Complexity:  

INSTALLATION

Static Data Available (0)

Controllers Installed (0)

Learning Ready (0)

All (0)

No applications in this state

LEARNING

Actively Learning (0)

Analysis Ready (1)

All (1)

STOP ALL MEASUREMENTS

Sort By: Most Recent 

 Assessment Hub Details

Stopped

 4d 6h 16m Time in Learning

[View Tech-Debt Report](#) 

ANALYSIS

Active Analysis (0)

Baseline Ready (2)

All (2)

Architectural Complexity  Sort By: Complexity High to Low 

 Medrec Demo 259 

Analysis Start- PB

7 Domains
99.97% Class Exclusivity

 OMS.NET 

Jeroen Domensino - 8 Dec 202...

8 Domains
86.31% Class Exclusivity



^ Filters Show All

Default View

Graph View

Date None

Week Month QTD

Status Show All None

TODO DONE Snoozed

Dismissed Acknowledged

Type Show All None

Drift Manual

Event Show All None

Architecture Complexity

Common Library Dead Code

Domain Changes

Dynamic Class Exclusivity

High Debt Class

Library Circular Dependencies

Resource Exclusivity

Static Class Exclusivity

Application Show All None

SEARCH

Assessment Hub Demo

Demo OMS Medrec Demo 259

OMS.NET

TODOS - 25 of 59 (4 Filters Applied)

GROUP BY: Application ▾



		Date	Event	Details	Domains
▼	● D	10-Sep-2023 12:55	Domain Changes	InventoryController	
▼	● D	10-Sep-2023 12:55	Domain Changes	OrderController (+1)	
▼	● D	10-Sep-2023 12:55	Domain Changes	StoreSearchController	
▼	● D	10-Sep-2023 12:55	Resource Exclusivity	oms-log.txt (File)	InventoryController, OrderController (+1), ShippingPriceController and 1 more.
▼	● D	27-Aug-2023 17:15	Architectural Complexity Decreased	Architectural complexity has decreased from 3 to 1	
▼	● D	25-Aug-2023 20:55	Architectural Complexity Decreased	Architectural complexity has decreased from 3 to 2	
▼	● D	26-Sep-2023 16:40	Domain Changes	ModifyFulfillmentController	
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	SALES_ORDER (Read-Write)	Application, ModifyFulfillmentController
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	SHIP_TO_ADDRESS (Read-Only)	Application, ModifyFulfillmentController
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	LINE_CHARGE (Read-Only)	Application, ModifyFulfillmentController
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	PAYMENT_INFO (Read-Only)	Application, ModifyFulfillmentController
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	BILL_TO_ADDRESS (Read-Only)	Application, ModifyFulfillmentController
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	CHARGES (Read-Only)	Application, ModifyFulfillmentController
▼	● D	26-Sep-2023 16:40	Resource Exclusivity	ORDER_LINE (Read-Write)	Application, ModifyFulfillmentController
▼	● D	27-Aug-2023 18:05	Architectural Complexity Increased	Architectural complexity has increased from 1 to 2	

^ Medrec Demo 259 Baseline measurement Analysis Completed

TODO | 4

DONE | 0

		Date	Event	Details	Domains
▼	● M	26-Oct-2023 08:55	Dynamic Class Exclusivity	PatientServiceImpl	Authentication, PatientSearch, RecordsManagement and 1 more.
▼	● M	18-Oct-2023 14:43	Dynamic Class Exclusivity	PatientService	Authentication, Monolith, RecordsManagement and 1 more.
▼	● D	3-Oct-2023 06:41	High Debt Class	JmsClientImpl	
▼	● D	3-Oct-2023 06:41	High Debt Class	PatientServiceImpl	

^ OMS.NET Baseline measurement Cori - 19 Jun 2023 12:56

TODO | 1

DONE | 3

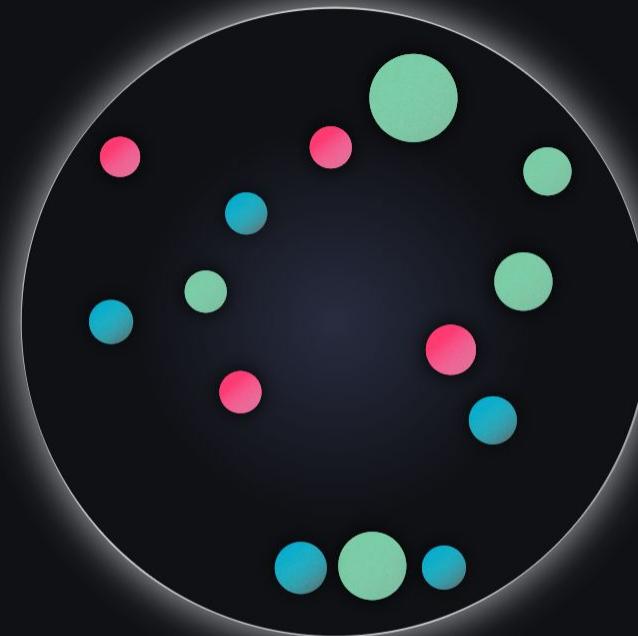
		Date	Event	Details	Domains
▼	● M	12-Oct-2023 14:09	Common Library	LineCharges	
▼	● M	15-Dec-2023 13:43	Dynamic Class Exclusivity	InventoryService	Found in 0 Domains
▼	● M	12-Oct-2023 14:11	Resource Exclusivity	dbo.OrderLines (Read-Write)	Found in 0 Domains
▼	● M	12-Oct-2023 14:06	Static Class Exclusivity	ShippingService	Found in 0 Domains



Learning

For the system to learn and understand your application, it needs to analyze as many usage patterns as possible. It is recommended to either collect data while the application is running in production for a while or, if it's installed in pre-production, test the major flows of the application. It is recommended to run your regular set of acceptance tests or regression tests in order to get the best coverage.

ACTIONS

[View Tech-Debt Report](#)[Select Controllers](#)

START

↗
MEASUREMENTS

14
DOMAINS

18,580 FUNCTIONS · 48,479 RESOURCES

Tech-Debt Report

[Cost of Innovation](#) [Tech Debt Metrics](#) [Aging Frameworks](#) [History](#)

The Tech-Debt Report shows the level of technical debt and architectural complexity in your application. Through static analysis, the report exposes how the dependencies between the classes affect the modularity of the code, the risk of impacting parts of the application and the overall level of technical debt. This level of debt affects the rate and cost of innovation and is a key business measure for the application.

This chart shows how the level of technical debt affects the cost of innovation for the application. It tries to assess if one would spend \$1 on this application, how much of it will actually be put to innovation, and how much to repay the technical debt within the application.

#Application Tags (0)



Pre-Refactor



Total cost of ownership multiplier

X7.7



87% - Technical debt

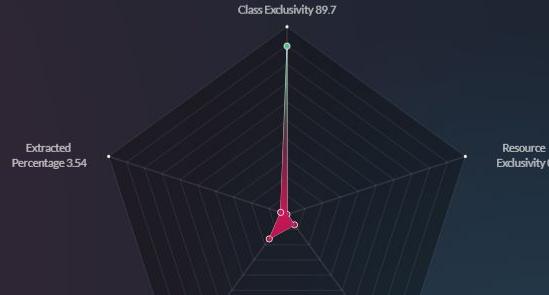


Architectural Complexity

The radar chart provides a visual representation of the key architectural aspects of your application. While these parameters do not indicate the complexity of the application, they do indicate the level of effort that is required to re-architect the application. A larger covered area signifies lower effort to re-architect the application.

This calculation is based on a specific measurement, and depends on the analysis that was done on it.

Complexity Score 5 Very high Effort



• 1
Low effort
0%-19%

• 2
Medium effort
20%-39%

• 3
Average effort
40%-59%

• 4
High effort

Tech-Debt Report

[Cost of Innovation](#) [Tech Debt Metrics](#) [Aging Frameworks](#) [History](#)

The Tech-Debt Report shows the level of technical debt and architectural complexity in your application. Through static analysis, the report exposes how the dependencies between the classes affect the modularity of the code, the risk of impacting parts of the application and the overall level of technical debt. This level of debt affects the rate and cost of innovation and is a key business measure for the application.

Complexity is the degree to which class dependencies are entangled between themselves, reducing the level of modularity of the code. Risk is correlated to the length of the dependencies. How likely is a change in the application to affect a seemingly unrelated part of it. The debt is an overall calculation, taking into account complexity, risk and other factors to provide a single metric to prioritize the application for modernization.

The average of these three metrics in the portfolio of applications is presented to further help with the prioritization.

#Application Tags (0)



Debt Index	
Score	86.54
Portfolio Avg	32
Difference	+168%

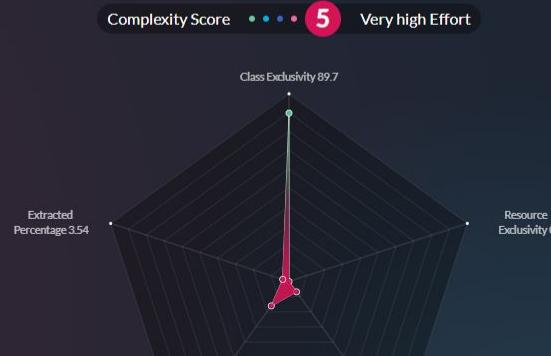
Risk Index	
Score	99
Portfolio Avg	32
Difference	+207%

Complexity Index	
Score	99
Portfolio Avg	31
Difference	+217%

Architectural Complexity

The radar chart provides a visual representation of the key architectural aspects of your application. While these parameters do not indicate the complexity of the application, they do indicate the level of effort that is required to re-architect the application. A larger covered area signifies lower effort to re-architect the application.

This calculation is based on a specific measurement, and depends on the analysis that was done on it.



Tech-Debt Report

An important part of technical debt is the aging frameworks that are used in the application. Aging frameworks are a source of security risk, and indicate an increased effort to upgrade the frameworks to their latest versions. The use of some aging frameworks will also block the application from being considered cloud-native.

This table shows the names of the jars of the frameworks that are used in the application.

Each framework, based on its current and newest version receives a tag of either modern, aging, or unknown, in case the specific jar was not found in the vFunction database.

It also shows the total number of classes that were scanned by the static analysis, and the minimal Java version that these classes were compiled with.

#Application Tags (0)



Aging Frameworks Threshold 0%			
Framework Validity • Modern • Unknown • Aging			
Name	Current Version	Last Version	Jar
Apache Ant Core	1.7.1	1.10.14	• ant-1.7.1.jar *
ANTLR 3 Tool	2.7.7	3.5.3	• antlr-2.7.7.jar *
ANTLR 4 Runtime	4.9.1	4.13.1	• antlr4-runtime-4.9.1.jar
Avalon Framework API	4.2.0	4.3.1	• avalon-framework-api-4.2...
Avalon Framework Imple...	4.2.0	4.3.1	• avalon-framework-impl-4...
Barcode4J	2.1	2.2.1	• barcode4j-2.1.jar
Bouncy Castle Provider	1.68	1.70.GM	• bcprov-jdk15on-1.68.jar *
bl	1.0.11-SNAPSHOT	5.1.0	• bl-1.0.11-SNAPSHOT.jar
bl	1.0.29-SNAPSHOT	5.1.0	• bl-1.0.29-SNAPSHOT.jar
nochia ant	1.1.1	1.7.1	• nochia_ant_1.1.1.jar *

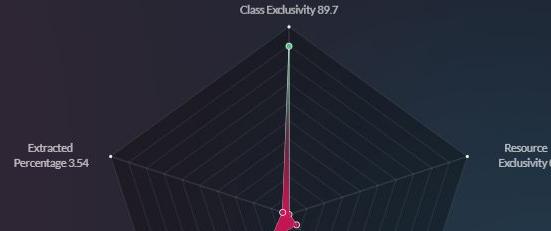


Architectural Complexity

The radar chart provides a visual representation of the key architectural aspects of your application. While these parameters do not indicate the complexity of the application, they do indicate the level of effort that is required to re-architect the application. A larger covered area signifies lower effort to re-architect the application.

This calculation is based on a specific measurement, and depends on the analysis that was done on it.

Complexity Score 5 Very high Effort



- 1 Low effort 0%-19%
- 2 Medium effort 20%-39%
- 3 Average effort 40%-59%

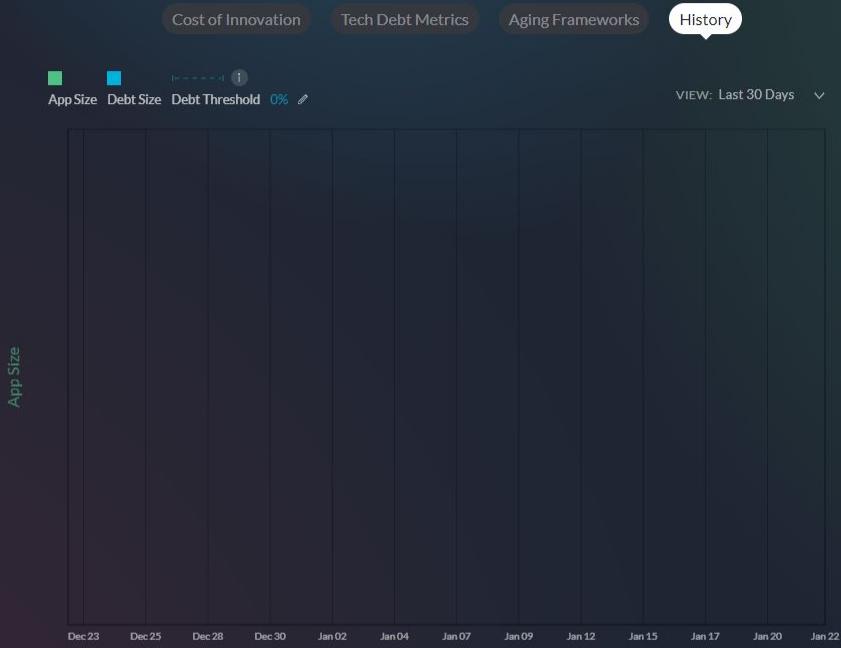
Tech-Debt Report

The size of the application and its technical debt are two key factors to consider when it comes to observability. The size of an application is related to the amount of code and resources that make up the application. A larger size can make it more difficult and costly to maintain and update. Technical debt, on the other hand, refers to the cost of maintaining and updating an application over time due to shortcuts or shortcuts taken during the development process. It can also refer to the cost of fixing problems or bugs that arise as a result of these shortcuts. A higher technical debt indicates that the application is more challenging to modernize.

The graph shows how the size and technical debt of the application changes over time. This information helps to understand the current state of the application and the potential challenges.

To help manage technical debt and ensure that it does not become too high, set a technical debt threshold for the application. A technical debt threshold is a specific level of technical debt that has been established as acceptable. The history graph's technical debt threshold tracks and monitors technical debt, and sends notifications when the technical debt of the application exceeds it. Staying aware of the technical debt of an application allows proactive measures to be taken to address it before it becomes a significant problem.

#Application Tags (0)



Architectural Complexity

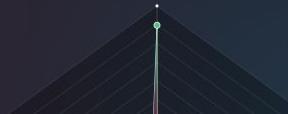
The radar chart provides a visual representation of the key architectural aspects of your application. While these parameters do not indicate the complexity of the application, they do indicate the level of effort that is required to re-architect the application. A larger covered area signifies lower effort to re-architect the application.

Complexity Score

5

Very high Effort

Class Exclusivity 89.7



- 1
Low effort
0%-19%
- 2
Medium effort
20%-39%

Cloud Native Innotation Labs

Step 3

Basculer sur l'instance locale de
vFunction et présenter MedRec
avant discovery + static analysis

Assess & Learn

Choose a controller to create an assessment report or to start the learning process.

The assessment report assesses the amount of technical debt within the application. It allows the user to determine the right modernization strategy and prioritize the applications accordingly.

The learning process enables vFunction to identify services and domains within the running application to support the design and extraction of services.

ACTIONS

 Install Controller

Filter By Status: UP WARNING DOWN

Filter By Created Date: Ascending ▾



IMPORT 

SET BASELINE 

CONFIGURATION

START LEARNING

VIEW ASSESSMENT REPORT

Hide Legend ▾

Controller Up

Attention Required

Controller Down

Assessment Report

[Cost of Innovation](#) [Tech Debt Metrics](#) [Aging Frameworks](#) [History](#)

The Assessment Report shows the level of technical debt and architectural complexity in your application. Through static analysis, the report exposes how the dependencies between the classes affect the modularity of the code, the risk of impacting parts of the application and the overall level of technical debt. This level of debt affects the rate and cost of innovation and is a key business measure for the application.

This chart shows how the level of technical debt affects the cost of innovation for the application. It tries to assess if one would spend \$1 on this application, how much it will actually be put to innovation, and how much to repay the technical debt within the application.

It also details the classes that contribute the most to the overall debt, by adding complexity and risk throughout the application. The post-refactor calculation assesses the level of technical debt assuming these classes were refactored, and their level of debt reduced.

Pre-Refactor



22% - Technical debt

Pre-Refactor
Total cost of ownership multiplier
x1.3

Post-Refactor
Total cost of ownership multiplier
x1.3

Refactoring Effort

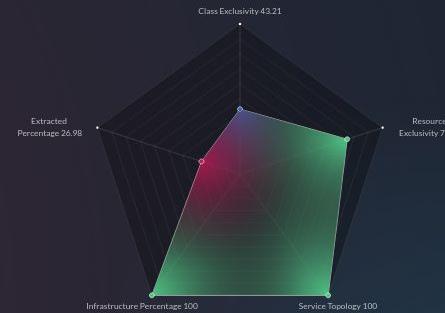
The radar chart shows the key modernization factors of the application. The larger the area is covered by the graph, the more the application is suited for modernization.

While these parameters do not directly indicate the complexity of the application itself, they indicate the level of effort that is required to re-architect the application.

This assessment is based on a specific measurement, and depends on the analysis that was done on it.

Enregistré dans ce PC

Assessment Score • 2 • • • Medium Effort



- 1 Low effort 0%-19%
- 2 Medium effort 20%-39%
- 3 Average effort 40%-59%
- 4 High effort 60%-79%
- 5 Very high effort 80%-100%

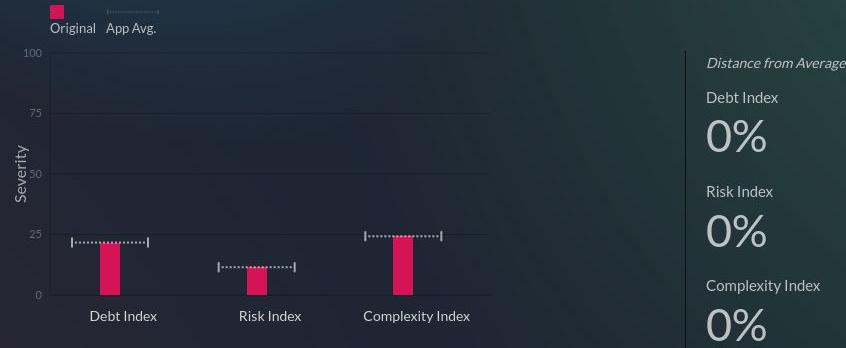
Assessment Report

Cost of Innovation Tech Debt Metrics Aging Frameworks History

The Assessment Report shows the level of technical debt and architectural complexity in your application. Through static analysis, the report exposes how the dependencies between the classes affect the modularity of the code, the risk of impacting parts of the application and the overall level of technical debt. This level of debt affects the rate and cost of innovation and is a key business measure for the application.

Complexity is the degree to which class dependencies are entangled between themselves, reducing the level of modularity of the code. Risk is correlated to the length of the dependencies. How likely is a change in the application to affect a seemingly unrelated part of it. The debt is an overall calculation, taking into account complexity, risk and other factors to provide a single metric to prioritize the application for modernization.

The average of these three metrics in the portfolio of applications is presented to further help with the prioritization.

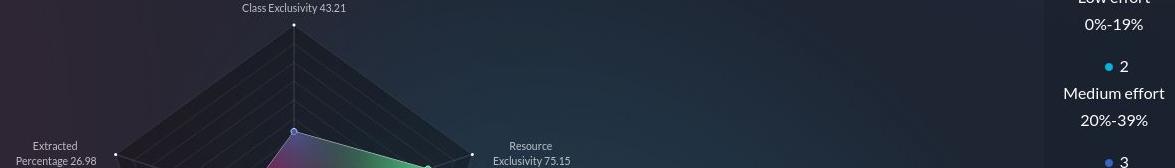


Refactoring Effort

Assessment Score • 2 • • • Medium Effort

The radar chart shows the key modernization factors of the application. The larger the area is covered by the graph, the more the application is suited for modernization.

While these parameters do not directly indicate the complexity of the application itself, they indicate the level of effort that is required to re-architect the application.



Cloud Native Innotation Labs

Step 4

Lancer une découverte et réaliser les 2 cas d'usage présentés + chat



Administrator

view and manage users.

The Administrator application allows an administrator to approve or deny new patient profile requests. The Administrator can reset passwords and create physicians.

[Login](#)[View Existing Users](#)

Patient

look up and view your visit and prescription history and edit your profile information.

The Patient application allows patients to log in, edit their profile information, or request that their profile be added to the system. Patients can also view prior medical records of visits with their physician.

[Login](#)[I'm New Here](#)

Physician

look up a patient, add visit information, and prescribe medications.

The Physician application allows physicians and nurses to log in, search and access patient profiles, create and review patient medical records, and prescribe medicine to patients.

[Go To The Physician App](#)



Administrator

view and manage users.
The Administrator application allows an administrator to approve or deny new patient profile requests. The Administrator can reset passwords and create physicians.

Login

View Existing Users

Administrator

x

Please sign in.

Email

admin@avitek.com

Password

●●●●●●●●

Sign In



Patient information

The Patient application allows patients to log in, search and access patient profiles, create and update their medical records, prescription history and edit your profile

Login

View Existing Users



Physician

look up a patient, add visit information, and prescribe medications.
The Physician application allows physicians and nurses to log in, search and access patient profiles, create and review patient medical records, and prescribe medicine to patients.

Go To The Physician App

Successful Login.

X

View pending patients

View pending applicant details and approve or deny them.

Go

Username	First Name	Middle Name	Last Name
charlie@star.com	Charlie	E	Florida
franck.besnard@devoteam.com	Franck		Besnard



Administrator

view and manage users.

The Administrator application allows an administrator to approve or deny new patient profile requests. The Administrator can reset passwords and create physicians.

[Login](#)[View Existing Users](#)

Patient

look up and view your visit and prescription history and edit your profile information.

The Patient application allows patients to log in, edit their profile information, or request that their profile be added to the system. Patients can also view prior medical records of visits with their physician.

[Login](#)[I'm New Here](#)

Physician

look up a patient, add visit information, and prescribe medications.

The Physician application allows physicians and nurses to log in, search and access patient profiles, create and review patient medical records, and prescribe medicine to patients.

[Go To The Physician App](#)

Email

Password

Physician

The Physician application allows physicians and nurses to log in, search and access patient profiles, create and review patient medical records, and prescribe medicine to patients.

There is an existing physician (mary@md.com / weblogic).

Successful Login.



Please input the patient's last name or SSN to find the patient you wish to diagnose.

When you input any character, the table below will show the patients who are consistent with the conditions of your query.

If the patient appears in the table, select the patient by clicking their last name.

Last Name

SSN

Last Name	First Name	Middle Name	DOB	SSN
Winner	Fred	I	Apr 26, 1975	123456789

Current Patient

First Name

Middle Name

Last Name

Fred

I

Winner

[Create Record](#)[View Patient](#)[Visit Records](#)[Prescriptions](#)

Date	Visit Reason	Physician
Aug 5, 2010	Twisted knee while playing soccer.	Mary J Oblige
Jul 18, 2009	Complains about chest pain.	Mary J Oblige
Jun 30, 2008	Sneezing, coughing, stuffy head.	Mary J Oblige
Jan 17, 2024	Sick	Mary J Oblige

[Oracle Home](#) | [Products and Services](#) | [Industries](#) | [Support](#) | [Store](#) | [Partners](#) | [Communities](#) | [About](#)Copyright © 2007, 2014, Oracle and/or its affiliates. All rights reserved. [Contact Oracle](#)

[View Record Summary](#) >> [View Record Detail](#)

Date Aug 5, 2010

Visit Reason Twisted knee while playing soccer.

Temperature 98

Weight 180

Pulse 75

Vital Signs

Height 70

Diastolic Blood Pressure 85

Systolic Blood Pressure 125

Exam Notes

Cast will be necessary before and after.

Diagnosis

Severely sprained interior ligament. Surgery required.

Prescriptions

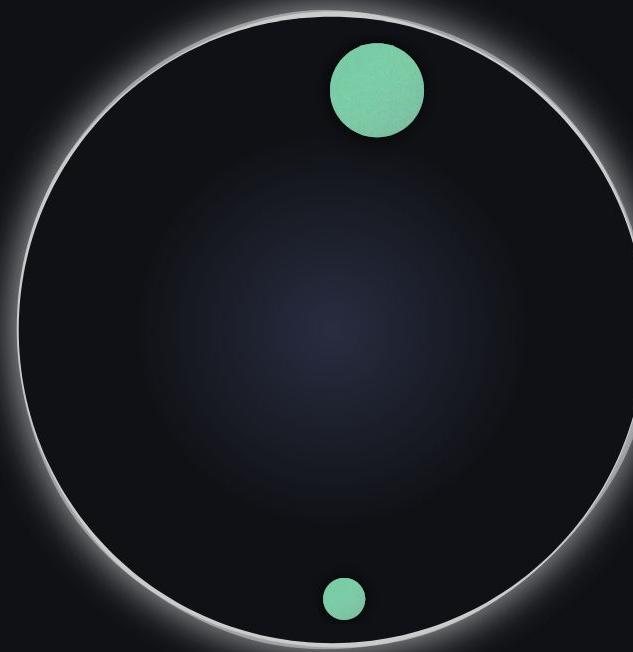
Date **Drug** **Dosage** **Frequency** **Refills** **Instructions**



Learning

For the system to learn your application, it needs to analyze as many usage patterns as possible. It is recommended to either collect data while the application is running in production for a while or, if it's installed in pre-production, test the functionality you are interested in splitting apart and containerizing. If you don't know what functionality to split, it is recommended to run your regular set of acceptance tests or regression tests.

ACTIONS

 [View Assessment Report](#) [Select Controllers](#) [Set Timer](#) [Download Measurement](#)

STOPPING

NEW MEASUREMENT

IMPORT

SELECT MEASUREMENT

SERVICES

2

FUNCTIONS

39

RESOURCES

27

Cloud Native Innotation Labs

Step 5

Charger une découverte complète,
basculer sur Analysis et présenter
les bubbles + focus sur le
PatientChatWebSocket (essayer
d'expliquer « au mieux » + préciser
que ce n'est pas magique !)



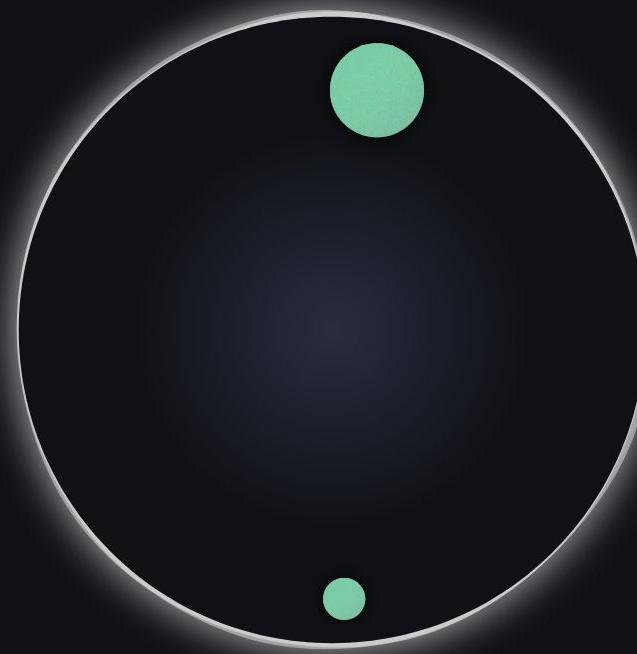
devoteam



Learning

For the system to learn your application, it needs to analyze as many usage patterns as possible. It is recommended to either collect data while the application is running in production for a while or, if it's installed in pre-production, test the functionality you are interested in splitting apart and containerizing. If you don't know what functionality to split, it is recommended to run your regular set of acceptance tests or regression tests.

ACTIONS

 [View Assessment Report](#) [Select Controllers](#) [Set Timer](#) [Download Measurement](#)

STOPPING

NEW MEASUREMENT

IMPORT

SELECT MEASUREMENT

SERVICES

2

FUNCTIONS

39

RESOURCES

27



SEARCH

SORT BY: Service name ▾ FILTERS (4)

AuthenticatingAdministratorController (6)

ViewingNewlyRegisteredPatientsCont (10) (+1)
roller.viewNewlyRegisteredPatients()

ViewingNewlyRegisteredPatientsController

Entry points EXPLORE TREE >

10 Dynamic classes Excl 70%

50 Static classes Excl 20%

3 Resources Excl 66%

0 Dead-code classes

Runtime 61.54%

SHOW DEPENDENCIES >

Monolith (6)

6 Dynamic classes Excl 66%

25 Static classes Excl 28%

25 Resources Excl 92%

0 Dead-code classes

Runtime 23.08%

Common Library (0)



Select measurement

- admin - 22 Jan 2024 11:49
- admin - 22 Jan 2024 10:15
- SNAP-002 Decoupling com...
- admin - 17 Jan 2024 18:42
- SNAP-001 Full discovery c...
- admin - 17 Jan 2024 18:41
- SNAP-000 Light discovery
- admin - 17 Jan 2024 18:30

ANALYSIS

44 MINUTES AGO

Services	2
Entry Points	4
Classes	19
Common Classes	0
Resource Exclusivity	92%
Class Exclusivity	84%
Extracted Percentage	68%
Time in Learning	23m
Ignored Classes	0
Total Classes	132

ACTIONS

View Assessment Report

Show History

Show Resources Report

Configure Parameters

Create Snapshot

Reset Analysis

Undo Last Action

Reset Graph

Reset Ignored Items

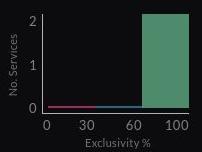
Download Measurement

▼

LEGEND

Size = No. of Classes

^



NEW MEASUREMENT

IMPORT

SELECT MEASUREMENT



SEARCH

SORT BY: Service name ▾ FILTERS (4)

● Address (1)

● ApprovingNewlyRegisteredPatientController (2)

● AuthenticatingAdministratorController (1)

● AuthenticationController.login() (2)

● BaseAuthenticationController (6)

● CreatingRecordController (4)

● JaxRsPhysicianFacadeBroker (3)

● JaxRsRecordFacadeBroker (+2) (6)

● JaxWsPatientFacadeBroker (2)

● PatientChatWebSocket (3)

PatientChatWebSocket

2 Entry points EXPLORE TREE >

3 Dynamic classes Excl 33%

0 Static classes Excl 0%

0 Resources Excl 0%

0 Dead-code classes

Runtime 0.86%

SHOW DEPENDENCIES >

● PatientNotifierBroker (3)

● PatientNotifierDelegate (1)

● PhysicianChatWebSocket (3)

● Record (1)

● SearchingPatientController (5)

● ViewingNewlyRegisteredPatientsController (2)

ANALYSIS 5 DAYS AGO

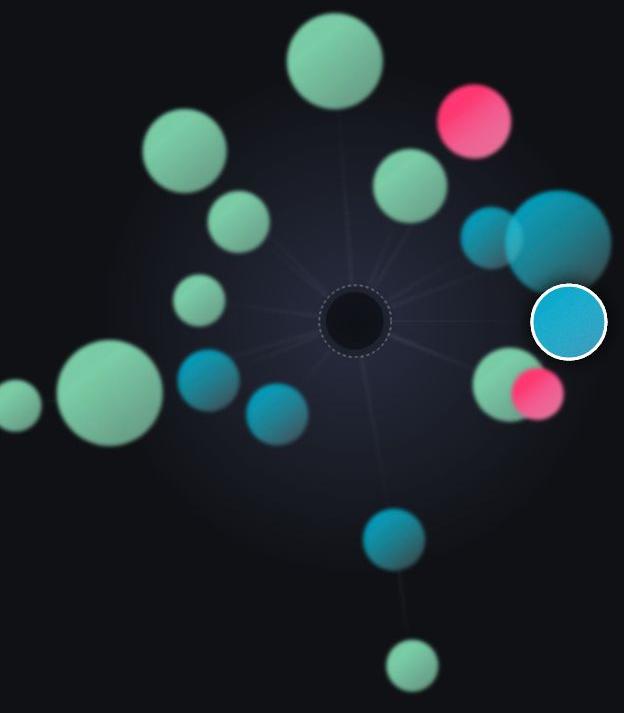
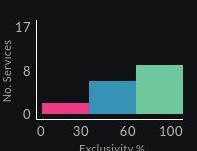
Services	17
Entry Points	30
Classes	47
Common Classes	5
Resource Exclusivity	89%
Class Exclusivity	82%
Extracted Percentage	71%
Ignored Classes	0
Total Classes	71

ACTIONS

- View Assessment Report
- Show History
- Show Resources Report
- Configure Parameters
- Create Snapshot
- Reset Analysis
- Undo Last Action
- Reset Graph
- Reset Ignored Items
- Download Measurement

LEGEND

Size = No. of Classes



NEW MEASUREMENT

IMPORT

SELECT MEASUREMENT



FILTER GROUP BY ▾

EXCL (1) NON (2) COMMON (0) ALL

ChatService (3)

com.oracle.medrec.chat.ChatService

_wl_cls_gen.jar

Not exclusive to the service

DETAILS >

Actions

IGNORE CLASS

MARK CLASS AS COMMON

MARK PACKAGE AS COMMON

MARK JAR AS INFRA

MAKE EXCLUSIVE

Show Methods ▾

ResponseMessageFactory (2)

com.oracle.medrec.chat.ResponseMessageFa
ctory

_wl_cls_gen.jar

Not exclusive to the service

DETAILS >

Actions

IGNORE CLASS

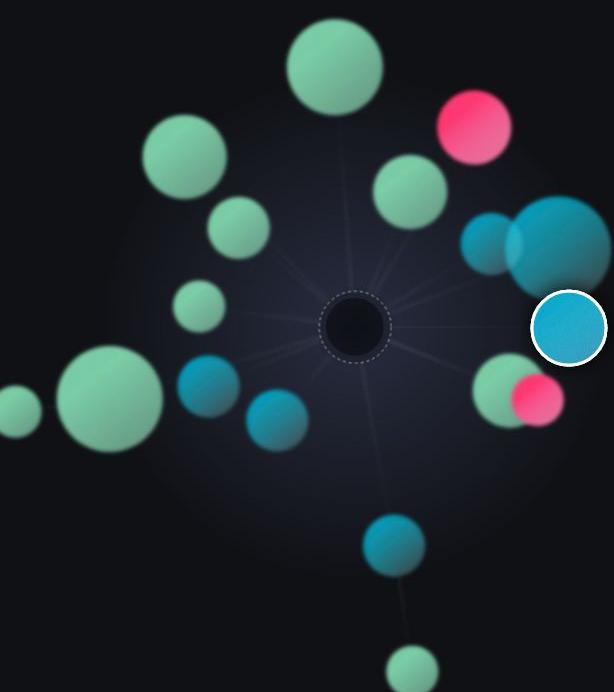
MARK CLASS AS COMMON

MARK PACKAGE AS COMMON

MARK JAR AS INFRA

MAKE EXCLUSIVE

Show Methods ▾



ANALYSIS 5 DAYS AGO

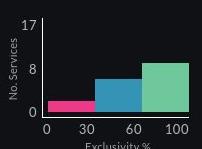
Services	17
Entry Points	30
Classes	47
Common Classes	5
Resource Exclusivity	89%
Class Exclusivity	82%
Extracted Percentage	71%
Ignored Classes	0
Total Classes	71

ACTIONS

- View Assessment Report
- Show History
- Show Resources Report
- Configure Parameters
- Create Snapshot
- Reset Analysis
- Undo Last Action
- Reset Graph
- Reset Ignored Items
- Download Measurement

LEGEND

Size = No. of Classes



NEW MEASUREMENT +

IMPORT +

SELECT MEASUREMENT ○



PatientChatWebSocket

FUNCTION STATES

<input type="radio"/> Third Party Functions	Hide	0
<input type="radio"/> Common Functions	Hide	0
<input type="radio"/> Other Functions		1
<input checked="" type="radio"/> Potential Entry Points		2
<input checked="" type="radio"/> Entry Points	Hide	1

ENTRY POINTS (2)

Merge

PatientChatWebSocket.onClose()

Shared the same variables or functions with, or called from, another entry point in the service

0 of 4 functions overlap
0 of 0 resources overlap

VIEW > MOVE REMOVE



PatientChatWebSocket.onMessage()

VIEW > MOVE REMOVE



PatientChatWebSocket

FUNCTION STATES

Third Party Functions	Hide	5
Common Functions	Hide	0
Other Functions		1
Potential Entry Points		2
Entry Points	Hide	12

ENTRY POINTS (2) Merge

PatientChatWebSocket.onClose()

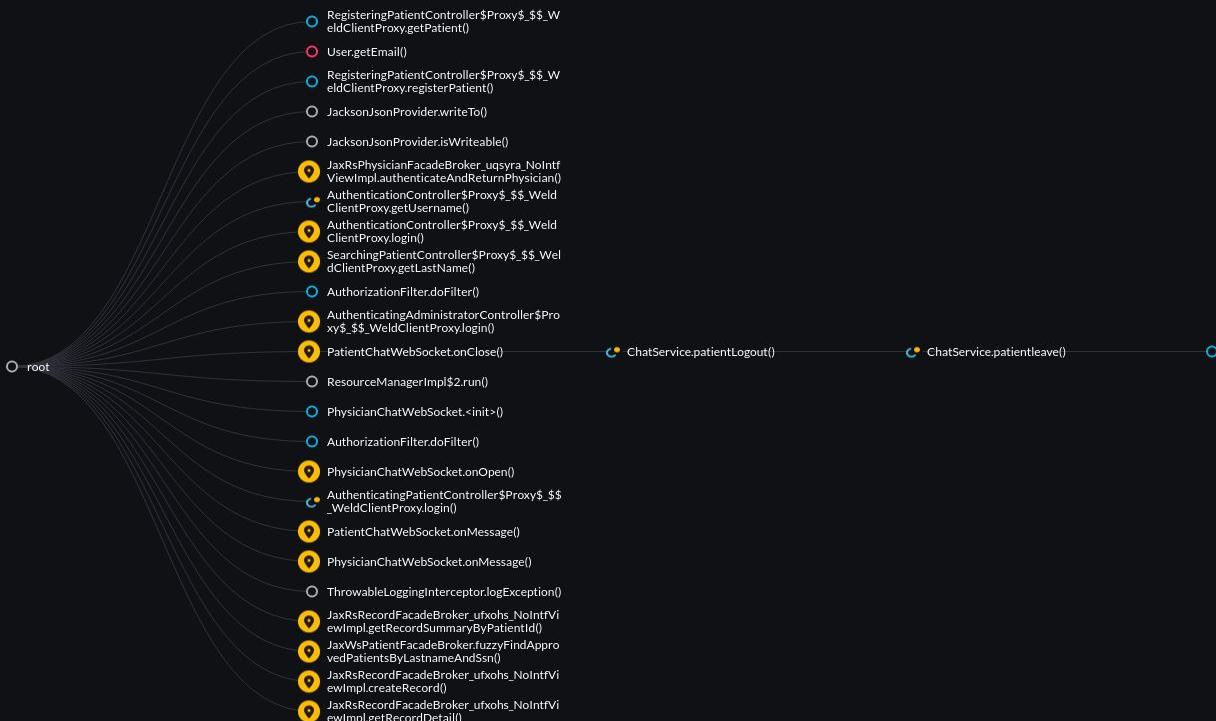
Shared the same variables or functions with, or called from, another entry point in the service

0 of 4 functions overlap
0 of 0 resources overlap

[VIEW](#) > [MOVE](#) [REMOVE](#)

PatientChatWebSocket.onMessage()

[VIEW](#) > [MOVE](#) [REMOVE](#)





SEARCH

SORT BY: Service name FILTERS (4)

- Address (1)
- ApprovingNewlyRegisteredPatientController (2)
- AuthenticatingAdministratorController (1)
- AuthenticationController.login() (2)
- BaseAuthenticationController (6)
- CreatingRecordController (4)
- JaxRsPhysicianFacadeBroker (3)
- JaxRsRecordFacadeBroker (+2) (6)
- JaxWsPatientFacadeBroker (2)
- PatientChatWebSocket (3)
- PatientNotifierBroker (3)
- PatientNotifierDelegate (1)
- PhysicianChatWebSocket (3)
- Record (1)
- SearchingPatientController (5)
- ViewingNewlyRegisteredPatientsController (2)
- ViewingPatientController (2)
- Monolith (11)
- Common Library (0)

ANALYSIS 5 DAYS AGO

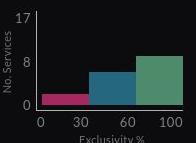
Services	17
Entry Points	30
Classes	47
Common Classes	5
Resource Exclusivity	89%
Class Exclusivity	82%
Extracted Percentage	71%
Ignored Classes	0
Total Classes	71

ACTIONS

- View Assessment Report
- Show History
- Show Resources Report
- Configure Parameters
- Create Snapshot
- Reset Analysis
- Undo Last Action
- Reset Graph
- Reset Ignored Items
- Download Measurement

LEGEND

Size = No. of Classes



Select measurement

- SNAP-001 Full discovery c...
- admin - 22 Jan 2024 11:49
- admin - 22 Jan 2024 10:15
- SNAP-002 Decoupling com...
- admin - 17 Jan 2024 18:42
- admin - 17 Jan 2024 18:41
- SNAP-000 Light discovery
- admin - 17 Jan 2024 18:30

NEW MEASUREMENT

IMPORT

SELECT MEASUREMENT

Cloud Native Innotation Labs

Step 6

Charger le SNAP-002 et préciser que l'objectif est de passer de la vue éclatée à la vue consolidée (préciser les compétences nécessaires + le jus de cerveau nécessaire) + extraction des codes sources des 3 webapps avec la target JEE 7



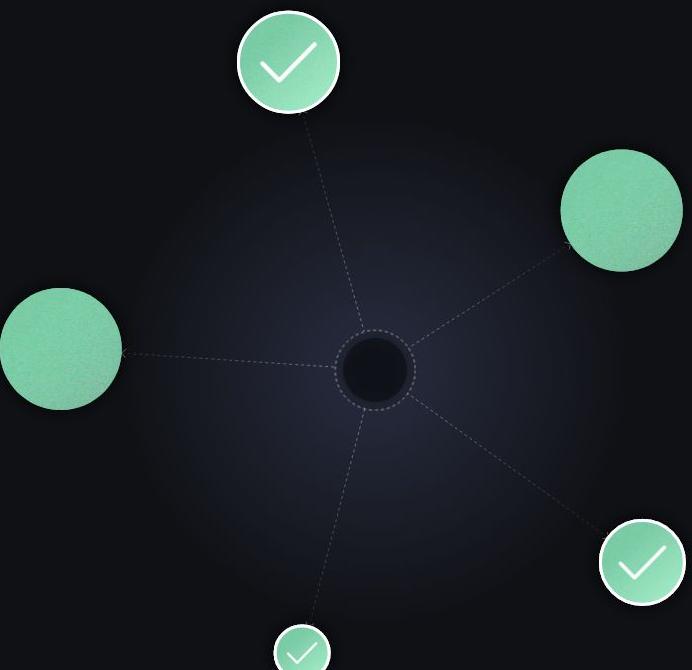


All Services

(5)

SEARCH

SORT BY: Service name ▾ FILTERS (4)

 Authentication * (13) DrPatientChat * (4) PatientSearch * (8) RecordsManagement * (13) RegisterNewPatients * (10) Monolith (1) Common Library (3)

ANALYSIS

ABOUT 3 HOURS AGO

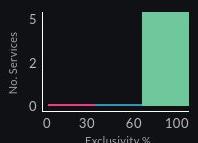
Services	5
Entry Points	28
Classes	42
Common Classes	6
Resource Exclusivity	93%
Class Exclusivity	92%
Extracted Percentage	89%
Ignored Classes	0
Total Classes	71

ACTIONS

- View Assessment Report
- Show History
- Show Resources Report
- Configure Parameters
- Create Snapshot
- Reset Analysis
- Undo Last Action
- Reset Graph
- Reset Ignored Items
- Download Measurement

LEGEND

Size = No. of Classes



NEW MEASUREMENT

IMPORT

SELECT MEASUREMENT



Service Creation

After analyzing your application, you can now automatically extract your new services.

Choose the services, download their service specification file, and then create your services by running the 'code-copy' tool. The 'code-copy' tool should run on a machine where you previously built the original application.

ACTIONS

- ⬇ Download Code-copy
- ⬇ Download Measurement

[CONFIGURE](#)[DOWNLOAD](#)

Authentication *	(13)
DrPatientChat *	(4)
PatientSearch *	(8)
RecordsManagement *	(13)
RegisterNewPatients *	(10)
Monolith	(1)
Common Library	(3)



Service Creation

After analyzing your application, you can now automatically extract your new services.

Choose the services, download their service specification file, and then create your services by running the 'code-copy' tool. The 'code-copy' tool should run on a machine where you previously built the original application.

ACTIONS

[Download Code-copy](#)[Download Measurement](#)



Service Configuration

Group ID	com.oracle.medrec.
Target Platform	JEE7
Target Version	JAVA_8
Dependency Repository Type	MAVEN

Generate endpoints

Download common library incrementally

[CANCEL](#) [SUBMIT](#)

[CONFIGURE](#)[DOWNLOAD](#)

- Authentication * (4)
- DrPatientChat * (4)
- PatientSearch * (8)
- RecordsManagement * (13)
- RegisterNewPatients * (10)
- Monolith (1)
- Common Library (3)



Service Creation

After analyzing your application, you can now automatically extract your new services.

Choose the services, download their service specification file, and then create your services by running the 'code-copy' tool. The 'code-copy' tool should run on a machine where you previously built the original application.

ACTIONS

- ⬇ Download Code-copy
- ⬇ Download Measurement

[CONFIGURE ✓](#)[DOWNLOAD](#)

● Authentication *	(13)
✓ DrPatientChat *	(4) ✕
✓ PatientSearch *	(8) ✕
● RecordsManagement *	(13)
✓ RegisterNewPatients *	(10) ✕
○ Monolith	(1)
✓ Common Library	(3)



Service Creation

After analyzing your application, you can now automatically extract your new services.

Choose the services, download their service specification file, and then create your services by running the 'code-copy' tool. The 'code-copy' tool should run on a machine where you previously built the original application.

ACTIONS

- [Download Code-copy](#)
- [Download Measurement](#)

Extract your first service

Download the code-copy tool [here](#).

Then provide the downloaded file to the tool.

See `code-copy --help` for instructions.

Do not show again

OK

CONFIGURE ✓

DOWNLOAD

CommonLibrary.json
Completed — 1.3 KB

RegisterNewPatients.yaml
Completed — 8.7 KB

RegisterNewPatients.json
Completed — 16.0 KB

DrPatientChat.yaml
Completed — 160 bytes

DrPatientChat.json
Completed — 2.4 KB

[Show all downloads](#)

Monolith

Common Library

ubuntu@developer: ~/Documents/Service Creation\$ more PatientSearch.json

```
{
  "additionalData": {
    "platform": "UNKNOWN",
    "root_conf_files": [],
    "entry_points": [
      {
        "class": "com.oracle.medrec.facade.jaxws.JaxWsPatientFacadeBroker",
        "method": "fuzzyFindApprovedPatientsByLastnameAndSsn",
        "is_static": false,
        "return": {
          "class": "java.util.List",
          "array_dimensions": 0,
          "type_args": [
            {
              "class": "com.oracle.medrec.facade.model.FoundPatient",
              "array_dimensions": 0,
              "type_args": null
            }
          ]
        },
        "args": [
          {
            "arg_name": "lastName",
            "arg_type": {
              "class": "java.lang.String",
              "array_dimensions": 0,
              "type_args": null
            },
            "arg_dir": "in"
          },
          {
            "arg_name": "ssn",
            "arg_type": {
              "class": "java.lang.String",
              "array_dimensions": 0,
              "type_args": null
            },
            "arg_dir": "in"
          }
        ],
        "class": "com.oracle.physician.web.controller.SearchingPatientController",
        "method": "search",
        "is_static": false,
        "return": {
          "class": "",
          "array_dimensions": 0,
          "type_args": null
        },
        "args": []
      },
      {
        "class": "com.oracle.physician.web.controller.SearchingPatientController",
        "method": "getLastNameInDTO",
        "is_static": false,
        "return": {
          "class": "com.oracle.medrec.facade.model.DTO",
          "array_dimensions": 0,
          "type_args": null
        },
        "args": []
      }
    ]
  }
}
```

ubuntu@developer: ~/Sandbox/medrec-refactored/patient-search\$ tree

```
.
├── build.gradle
├── pom.xml
└── pom.xml.orig
    └── settings.gradle
    └── src
        ├── main
        │   └── java
        │       └── com
        │           └── oracle
        │               └── medrec
        │                   └── facade
        │                       ├── PatientFacade.java
        │                       └── broker
        │                           └── jaxws
        │                               └── JaxWsPatientFacadeBroker.java
        │
        └── impl
            └── PatientFacadeImpl.java
        └── patientsearch
            ├── JaxRSConfig.java
            ├── JaxWsPatientFacadeBrokerController.java
            └── SearchingPatientControllerController.java
            └── dto
                └── JaxWsPatientFacadeBrokerFuzzyFindApprovedPatientsByLastnameAndSsnInD
                    └── JaxWsPatientFacadeBrokerFuzzyFindApprovedPatientsByLastnameAndSsnOut
                        ├── SearchingPatientControllerGetLastNameInDTO.java
                        ├── SearchingPatientControllerGetLastNameOutDTO.java
                        └── SearchingPatientControllerSearchInDTO.java
                        └── SearchingPatientControllerSearchOutDTO.java
            └── service
                └── PatientService.java
                └── impl
                    └── PatientServiceImpl.java
            └── physician
                └── JaxWSProperties.java
            └── service
                └── PatientService.java
                └── delegate
                    └── PatientServiceDelegate.java
                    └── converter
                        └── PatientConverter.java
                        └── impl
                            └── PatientConverterImpl.java
                            └── RegularUserConverterImpl.java
            └── stub
                └── jaxws
                    ├── Address.java
                    ├── BaseEntity.java
                    └── DomainModel.java
                    └── FindApprovedPatientBySsn.java
                    └── FindApprovedPatientBySsnResponse.java
                    └── FindApprovedPatientsByLastName.java
                    └── FindApprovedPatientsByLastNameResponse.java
```

Cloud Native Innotation Labs

Step 7

On bascule sur le studio Eclipse +
explication de comment c'est
obtenu + montrer rapidement les
fichiers modifiés pour le service
PatientSearch



devoteam

File Edit Source Navigate Search Project Run Window Help



Package Explorer X

- commonlib
- DrPatientChat
- PatientSearch
 - src/main/java
 - com.oracle.medrec.facade
 - com.oracle.medrec.facade.broker.jaxws
 - com.oracle.medrec.facade.impl
 - com.oracle.medrec.patientsearch
 - com.oracle.medrec.patientsearch.dto
 - com.oracle.medrec.service
 - com.oracle.medrec.service.impl
 - com.oracle.physician
 - com.oracle.physician.service
 - com.oracle.physician.service.delegate
 - com.oracle.physician.service.delegate.converter
 - com.oracle.physician.service.delegate.converter.ir
 - com.oracle.physician.service.delegate.stub.jaxws
 - com.oracle.physician.web.controller
 - src/main/resources
 - src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- build.gradle
- pom.xml
- pom.xml.org
- settings.gradle
- RegisterNewPatients

PatientSearch/pom. x persistence.xml web.xml JaxRSConfig.java PatientServiceImpl. PatientService.java BasePhysicianPageC

```
<?xml version="1.0" encoding="UTF-8"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd" xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <!-- artifact coordinates -->
  <groupId>com.oracle.medrec</groupId>
  <artifactId>PatientSearch</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <!-- properties copied from the parent pom.xml file referenced by the original medrec code -->
  <!-- please review the below and ensure it is set correctly to your system -->
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <wls.url>t3://10.0.0.3:7001</wls.url>
    <wls.user>weblogic</wls.user>
    <wls.password>welcome1</wls.password>
    <wls.adminServer>AdminServer</wls.adminServer>
  </properties>
  <build>
    <defaultGoal>install</defaultGoal>
    <finalName>${project.name}</finalName>
    <pluginManagement>
      <!-- plugins copied from the parent pom.xml file referenced by the original medrec code -->
      <plugins>
        <plugin>
          <!-- This is the configuration for the weblogic-maven-plugin -->
          <groupId>com.oracle.weblogic</groupId>
          <artifactId>weblogic-maven-plugin</artifactId>
          <version>12.1.3-0-0</version>
          <configuration>
            <adminurl>${wls.url}</adminurl>
            <user>${wls.user}</user>
            <password>${wls.password}</password>
            <!--The location of the file or directory to be deployed -->
            <source>${project.build.directory}/${project.build.finalName}.${project.packaging}</source>
            <!--The target servers where the application is deployed. -->
          </configuration>
        
```

Outl X

Grammars

jar:file:/snapshots/Binding.xsd Cache: fail

xml project

Problems X Javadoc Declaration

0 errors, 46 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (46 items)				

Writable Insert 1 : 1 : 0 LemMinX: (15%)

File Edit Source Navigate Search Project Run Window Help

Package Explorer X

- commonlib
- DrPatientChat
- PatientSearch
 - src/main/java
 - com.oracle.medrec.facade
 - com.oracle.medrec.facade.broker.jaxws
 - com.oracle.medrec.facade.impl
 - com.oracle.medrec.patientsearch
 - com.oracle.medrec.patientsearch.dto
 - com.oracle.medrec.service
 - com.oracle.medrec.service.impl
 - com.oracle.physician
 - com.oracle.physician.service
 - Enregistré dans ce PC
 - com.oracle.physician.service.delegate
 - com.oracle.physician.service.delegate.converter
 - com.oracle.physician.service.delegate.converter.ir
 - com.oracle.physician.service.delegate.stub.jaxws
 - com.oracle.physician.web.controller
 - src/main/resources
 - src/test/java
 - JRE System Library [javaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - build.gradle
 - pom.xml
 - pom.xml.orig
 - settings.gradle

RegisterNewPatients

http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd (xsi:schemaLocation)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
3     <persistence-unit name="MyMedRec" transaction-type="JTA">
4         <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
5         <jta-data-source>jdbc/MedRecGlobalDataSourceXA</jta-data-source>
6         <class>com.oracle.medrec.model.Address</class>
7         <class>com.oracle.medrec.model.BaseEntity</class>
8         <class>com.oracle.medrec.model.PersonName</class>
9         <class>com.oracle.medrec.model.Patient</class>
10        <class>com.oracle.medrec.model.Physician</class>
11        <class>com.oracle.medrec.model.Prescription</class>
12        <class>com.oracle.medrec.model.Record</class>
13        <class>com.oracle.medrec.model.RegularUser</class>
14        <class>com.oracle.medrec.model.User</class>
15        <class>com.oracle.medrec.model.VersionedEntity</class>
16        <class>com.oracle.medrec.model.VitalSigns</class>
17        <properties>
18            <property name="eclipselink.logging.level" value="FINE"/>
19        </properties>
20    </persistence-unit>
21 </persistence>
```

Task X

Find All A

Outl X

Grammars

http://java.su Binding X

Cache file:///ho xml persistence

Problems X Javadoc Declaration

0 errors, 47 warnings, 0 others

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

► Warnings (47 items)

File Edit Source Navigate Search Project Run Window Help

Package Explorer X

- commonlib
- DrPatientChat
- PatientSearch
 - src/main/java
 - com.oracle.medrec.facade
 - com.oracle.medrec.facade.broker.jaxws
 - com.oracle.medrec.facade.impl
 - com.oracle.medrec.patientsearch
 - com.oracle.medrec.patientsearch.dto
 - com.oracle.medrec.service
 - com.oracle.medrec.service.impl
 - com.oracle.physician
 - com.oracle.physician.service
 - com.oracle.physician.service.delegate
 - com.oracle.physician.service.delegate.converter
 - com.oracle.physician.service.delegate.converter.ir
 - com.oracle.physician.service.delegate.stub.jaxws
 - com.oracle.physician.web.controller
 - src/main/resources
 - src/test/java
 - JRE System Library [javaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - build.gradle
 - pom.xml
 - pom.xml.orig
 - settings.gradle

RegisterNewPatients

PatientSearch/pom. persistence.xml web.xml X JaxRSConfig.java PatientServiceImpl. PatientService.java BasePhysicianPageC

```
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd (xsi:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
5 </web-app>
6
```

Task X

Find All A

Outl X

Grammars

http://java.s...

- Binding: x
- Cache
- file:///h...
- xml
- web-app

Problems X Javadoc Declaration

0 errors, 47 warnings, 0 others

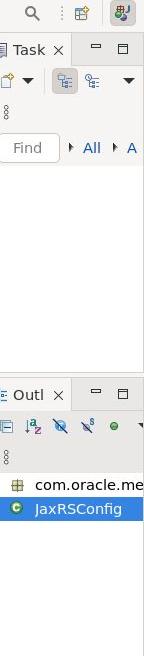
Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

► Warnings (47 items)

File Edit Source Refactor Navigate Search Project Run Window Help



```
1 package com.oracle.medrec.patientsearch;
2
3 import javax.ws.rs.ApplicationPath;
4
5 @ApplicationPath("/")
6 public class JaxRSConfig extends Application {
7 }
8 }
```



Problems X Javadoc Declaration

0 errors, 47 warnings, 0 others

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

▶ Warnings (47 items)

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- commonlib
- DrPatientChat
- PatientSearch
 - src/main/java
 - com.oracle.medrec.facade
 - com.oracle.medrec.facade.broker.jaxws
 - com.oracle.medrec.facade.impl
 - com.oracle.medrec.patientsearch
 - com.oracle.medrec.patientsearch.dto
 - com.oracle.medrec.service
 - com.oracle.medrec.service.impl
 - com.oracle.physician
 - com.oracle.physician.service
 - com.oracle.physician.service.delegate
 - com.oracle.physician.service.delegate.converter
 - com.oracle.physician.service.delegate.converter.ir
 - com.oracle.physician.service.delegate.stub.jaxws
 - com.oracle.physician.web.controller
 - src/main/resources
 - src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- build.gradle
- pom.xml
- pom.xml.org
- settings.gradle

RegisterNewPatients

```
1 package com.oracle.medrec.service.impl;
2
3 import com.oracle.medrec.common.persistence.CriteriaPersistenceSupport;
4
5 /**
6  * A patient business service implementation which is responsible for all
7  * business operations to patient.
8  *
9  * @author Xiaojun Wu. <br>
10 * Copyright (c) 2007, 2014, Oracle and/or its
11 * affiliates. All rights reserved.
12 */
13 @Stateless
14 public class PatientServiceImpl extends BaseUserServiceImpl<Patient> implements
15     PatientService {
16
17     /* @EJB(beanName = "PatientNotifierDelegate")
18     private PatientNotifier patientNotifier;
19
20     public void createPatient(Patient patient)
21         throws DuplicateUsernameException, DuplicateSsnException {
22         isDuplicateSsn(patient);
23         super.createUser(patient);
24     }
25
26     public Patient getPatient(Long patientId) {
27         return entityManager.find(entityClass, patientId);
28     }
29
30     public Patient findApprovedPatientBySsn(String ssn) {
31         return CriteriaPersistenceSupport.findUnique(entityManager,
32             criteriaBuilder, entityClass, PredicationFactory
33                 .createEqualPredication(ssn, "ssn"), PredicationFactory
34                 .createEqualPredication(Patient.Status.APPROVED,
35                     "status"));
36     }
37
38     @TransactionAttribute(TransactionAttributeType.NOT_SUPPORTED)
39 }
```

Task X

Find All A

Outl X

com.oracle.me

- PatientService
 - getPatient(L)
 - findApprovedPatientBySsn(S)
 - fuzzyFindApp

Problems X @ Javadoc Declaration

0 errors, 47 warnings, 0 others

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

► Warnings (47 items)

Writable

Smart Insert

1 : 1 : 0

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- commonlib
- DrPatientChat
- PatientSearch
 - src/main/java
 - com.oracle.medrec.facade
 - com.oracle.medrec.facade.broker.jaxws
 - com.oracle.medrec.facade.impl
 - com.oracle.medrec.patientsearch
 - com.oracle.medrec.patientsearch.dto
 - com.oracle.medrec.service
 - com.oracle.medrec.service.impl
 - com.oracle.physician
 - com.oracle.physician.service
 - com.oracle.physician.service.delegate
 - com.oracle.physician.service.delegate.converter
 - com.oracle.physician.service.delegate.converter.ir
 - com.oracle.physician.service.delegate.stub.jaxws
 - com.oracle.physician.web.controller
 - src/main/resources
 - src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- build.gradle
- pom.xml
- pom.xml.orig
- settings.gradle

RegisterNewPatients

PatientSearch/pom.xml persistence.xml web.xml JaxRSConfig.java PatientServiceImpl.java PatientService.java BasePhysicianPageC

```
1 package com.oracle.physician.service;
2
3 import com.oracle.medrec.facade.PatientFacade;
4
5 /**
6  * That this service interface directly inherit from the medrec facade interface
7  * is just for simplicity here, since we don't want to re-create another set of
8  * interfaces and models in this sample app. For real-world apps, (webservices)
9  * clients are totally decoupled from server side, and should have their object
10 * models (maybe in different language).
11 *
12 * @author Copyright (c) 2007, 2014, Oracle and/or its
13 * affiliates. All rights reserved.
14 */
15 public interface PatientService extends PatientFacade {
16
17 }
18
```

Problems X Javadoc Declaration

0 errors, 47 warnings, 0 others

Description	Resource	Path	Location	Type
▶ Warnings (47 items)				

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X

- commonlib
- DrPatientChat
- PatientSearch
 - src/main/java
 - com.oracle.medrec.facade
 - com.oracle.medrec.facade.broker.jaxws
 - com.oracle.medrec.facade.impl
 - com.oracle.medrec.patientsearch
 - com.oracle.medrec.patientsearch.dto
 - com.oracle.medrec.service
 - com.oracle.medrec.service.impl
 - com.oracle.physician
 - com.oracle.physician.service
 - com.oracle.physician.service.delegate
 - com.oracle.physician.service.delegate.converter
 - com.oracle.physician.service.delegate.converter.converter
 - com.oracle.physician.service.delegate.stub.jaxws
 - com.oracle.physician.web.controller
 - src/main/resources
 - src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- build.gradle
- pom.xml
- pom.xml.orig
- settings.gradle

RegisterNewPatients

PatientSearch/pom.xml persistence.xml web.xml JaxRSConfig.java PatientServiceImpl.java PatientService.java BasePhysicianPageC X

```
1 package com.oracle.physician.web.controller;
2
3 import com.oracle.physician.common.web.PageControllerSupport;
4
5 /**
6  * @author Copyright (c) 2007, 2014, Oracle and/or its
7  * affiliates. All rights reserved.
8  */
9 public abstract class BasePhysicianPageController extends PageControllerSupport {
10
11     /* @author Copyright (c) 2007, 2014, Oracle and/or its
12     * affiliates. All rights reserved.
13     */
14
15     /**
16      * @Inject
17      * private PhysicianService physicianService;
18
19      * @Inject
20      * private PatientService patientService;
21
22      * @Inject
23      * private RecordService recordService;
24
25      * protected PhysicianService getPhysicianService() {
26      *     return physicianService;
27      * }
28
29      * protected PatientService getPatientService() {
30      *     return patientService;
31      * }
32
33      * protected RecordService getRecordService() {
34      *     return recordService;
35      * }
36
37 }
```

Problems X Javadoc Declaration

0 errors, 47 warnings, 0 others

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------

► Warnings (47 items)

Writable

Smart Insert

1 : 1 : 0

```
ubuntu@developer: ~/Sandbox/medrec-refactored/patient-search [ ]  
e declaration of plugin org.apache.maven.plugins:maven-compiler-plugin @ line 89, column 12  
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your bui-  
ld.  
[WARNING] For this reason, future Maven versions might no longer support building such malformed project  
s.  
[WARNING]  
[INFO]  
[INFO] -----< com.oracle.medrec:PatientSearch >-----  
[INFO] Building PatientSearch 1.0-SNAPSHOT  
[INFO] -----[ war ]-----  
[INFO]  
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ PatientSearch ---  
[INFO] Deleting /home/ubuntu/Sandbox/medrec-refactored/patient-search/target  
[INFO]  
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ PatientSearch ---  
[INFO] Using 'UTF-8' encoding to copy filtered resources.  
[INFO] Copying 3 resources  
[INFO]  
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ PatientSearch ---  
[INFO] Changes detected - recompiling the module!  
[INFO] Compiling 48 source files to /home/ubuntu/Sandbox/medrec-refactored/patient-search/target/classes  
[INFO]  
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ PatientSearch ---  
[INFO] Not copying test resources  
[INFO]  
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ PatientSearch ---  
[INFO] Not compiling test sources  
[INFO]  
[INFO] --- maven-surefire-plugin:2.15:test (default-test) @ PatientSearch ---  
[INFO] Tests are skipped.  
[INFO]  
[INFO] --- maven-war-plugin:2.2:war (default-war) @ PatientSearch ---  
[INFO] Packaging webapp  
[INFO] Assembling webapp [PatientSearch] in [/home/ubuntu/Sandbox/medrec-refactored/patient-search/targe-  
t/PatientSearch]  
[INFO] Processing war project  
[INFO] Copying webapp resources [/home/ubuntu/Sandbox/medrec-refactored/patient-search/src/main/webapp]  
[INFO] Webapp assembled in [171 msec(s)]  
[INFO] Building war: /home/ubuntu/Sandbox/medrec-refactored/patient-search/target/PatientSearch.war  
[INFO] WEB-INF/web.xml already added, skipping  
[INFO]  
[INFO] --- maven-install-plugin:2.4:install (default-install) @ PatientSearch ---  
[INFO] Installing /home/ubuntu/Sandbox/medrec-refactored/patient-search/target/PatientSearch.war to /hom-  
e/ubuntu/.m2/repository/com/oracle/medrec/PatientSearch/1.0-SNAPSHOT/PatientSearch-1.0-SNAPSHOT.war  
[INFO] Installing /home/ubuntu/Sandbox/medrec-refactored/patient-search/pom.xml to /home/ubuntu/.m2/repo-  
sitory/com/oracle/medrec/PatientSearch/1.0-SNAPSHOT/PatientSearch-1.0-SNAPSHOT.pom  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO]  
[INFO] -----  
[INFO] Total time: 7.302 s  
[INFO] Finished at: 2024-01-22T13:32:15Z  
[INFO] -----  
ubuntu@developer:~/Sandbox/medrec-refactored/patient-search$ [ ]
```

```
ubuntu@developer: ~/Sandbox/medrec-refactored/artifacts [ ]  
total 5216  
-rw-rw-r-- 1 ubuntu ubuntu 11891 Jan 15 15:49 DrPatientChat.war  
-rw-rw-r-- 1 ubuntu ubuntu 2662939 Jan 15 17:06 PatientSearch.war  
-rw-rw-r-- 1 ubuntu ubuntu 2662148 Jan 15 19:03 RegisterNewPatients.war  
ubuntu@developer:~/Sandbox/medrec-refactored/artifacts$ [ ]
```

Cloud Native Innotation Labs

Step 8

On termine en présentant la création
des containers et en montrant le
test des APIs REST

Container list ⚙

 Containers weblog

+ Add container



<input type="checkbox"/>	Name ↓	State ↑ Filter ▾	Quick Actions	Stack ↓↑	Image ↗	Created ↓↑	IP Address ↓↑	Published Ports ↑	Ownership ↓↑
<input type="checkbox"/>	patientChat	healthy		-	besn0847/weblogic-base:12	2024-01-20 10:08:15	172.17.0.4	10003:7001	administrators
<input type="checkbox"/>	patientSearch	healthy		-	besn0847/weblogic-base:12	2024-01-20 10:10:00	172.17.0.2	10001:7001	administrators
<input type="checkbox"/>	registerNewPatient	healthy		-	besn0847/weblogic-base:12	2024-01-20 10:10:19	172.17.0.3	10002:7001	administrators

Items per page

Home

local

Dashboard

App Templates

Stacks

Containers

Images

Networks

Volumes

Events

Host

Settings

Users

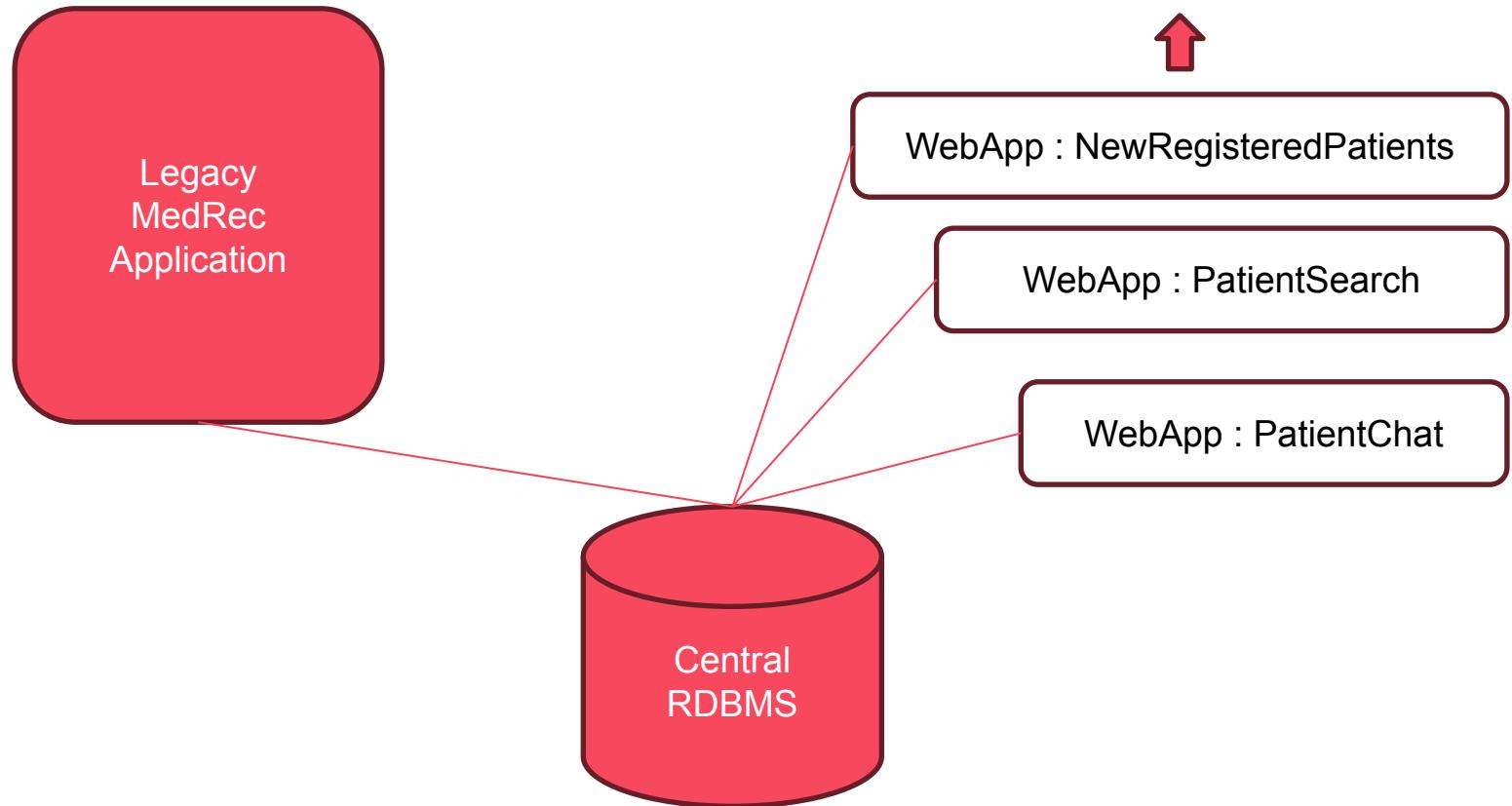
Environments

Registries

Authentication logs

Notifications

Settings



12^c

Welcome

Log in to work with the WebLogic Server domain

Username:

Password:

ORACLE WebLogic Server Administration Console 12c

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

- base_domain
 - Domain Partitions
 - Environment
 - Deployments
 - Services
 - Security Realms
 - Interoperability
 - Diagnostics

How do I...

- Configure testing options for a JDBC data source
- Configure the statement cache for a JDBC connection pool
- Configure credential mapping for a JDBC data source
- Configure connection harvesting for a connection pool
- Encrypt connection properties

System Status

Health of Running Servers as of 1:46 PM

Failed (0)	Critical (0)
Overloaded (0)	Warning (0)
OK (1)	

Settings for MedRecGlobalDataSourceXA

Configuration Targets Monitoring Control Security Notes

General Connection Pool Transaction Diagnostics Identity Options

Save

The connection pool within a JDBC data source contains a group of JDBC connections that applications reserve, use, and then return to the pool. The connection pool and the connections within it are created when the connection pool is registered, usually when starting up WebLogic Server when deploying the data source to a new target.

Use this page to define the configuration for this data source's connection pool.

URL: jdbc:derby://dbserver:1527/medrec;ServerName=dbserver;database=medrec

The URL of the database to connect to. The format of the URL varies by JDBC driver. [More Info...](#)

Driver Class Name: org.apache.derby.jdbc.ClientXADataSource

The full package name of JDBC driver class used to create the physical database connections in the connection pool. (Note that this driver class must be in the classpath of any server to which it is deployed.) [More Info...](#)

Properties:

```
user=medrec
portNumber=1527
databaseName=medrec;create=true
serverName=dbserver
```

The list of properties passed to the JDBC driver that are used to create physical database connections. For example: server=dbserver1. List each property=value pair on a separate line. [More Info...](#)

System Properties:

The list of system properties names passed to the JDBC driver that are used to create physical database connections. For example: server=dbserver1. List each property=value pair on a separate line. [More Info...](#)

Encrypted Properties:

Add Securely

The list of encrypted properties passed to the JDBC driver that are used to create physical database connections. For example: password=value. [More Info...](#)

Password:

The password attribute passed to the JDBC driver when creating physical database connections. [More Info...](#)

Confirm Password:

Initial Capacity: 1

The number of physical connections to create when creating the connection pool in the data source. If unable to create this number of connections, creation of the data source will fail. [More Info...](#)

Maximum Capacity: 15

The maximum number of physical connections that this connection pool can contain. [More Info...](#)

Minimum Capacity: 1

The minimum number of physical connections that this connection pool can contain after it is initialized. [More Info...](#)

Statement Cache Type: LRU

The algorithm used for maintaining the prepared statements stored in the statement cache. [More Info...](#)

12.0.0.10:10001/console/console.portal?_nfpb=true&_pageLabel=ChangeManagementPage

Welcome, weblogic | Connected to: base_domain

ORACLE WebLogic Server Administration Console 12c

[Home](#) Log Out Preferences [Record](#) Help

Welcome, weblogic | Connected to: **base_domain**

Change Center

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure

- base_domain
 - + Domain Partitions
 - + Environment
 - Deployments**
 - + Services
 - Security Realms
 - + Interoperability
 - + Diagnostics

How do I...

- Install an enterprise application
- Configure an enterprise application
- Update (redeploy) an enterprise application
- Monitor the modules of an enterprise application
- Deploy EJB modules
- Install a Web application

System Status

Health of Running Servers as of 1:50 PM

Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
OK (1)

Summary of Deployments

Configuration Control Monitoring

This page displays the list of Java EE applications and standalone application modules installed to this domain. You can update (redeploy) or delete installed applications and modules from the domain by selecting the checkbox next to the application name and then using the controls on this page. To install a new application or module for deployment to targets in this domain, click **Install**.

Customize this table

Deployments

Showing 1 to 1 of 1 Previous | Next

<input type="checkbox"/> Name	State	Health	Type	Targets	Scope	Domain Partitions	Deployment Order
<input type="checkbox"/> PatientSearch	Active	✓ OK	Web Application	AdminServer	Global		100
EJBs							
CachingNamingClientDecorator			EJB				
JaxWsPatientFacadeBrokerController			EJB				
JmsClientImpl			EJB				
MailClientImpl			EJB				
MessageComposerImpl			EJB				
MethodInvocationCacheImpl			EJB				
MethodParameterValidatorImpl			EJB				
NamingClientImpl			EJB				
PatientConverterImpl			EJB				
PatientFacadeImpl			EJB				
PatientServiceImpl			EJB				
SearchingPatientControllerController			EJB				
ThrowableLoggerImpl			EJB				
Web Services							
None to display							
REST Services							
[*] com.oracle.medrec.patientsearch.JaxRSConfig com.oracle.medrec.patientsearch.JaxRSConfig			REST Service				

Install Update Delete

Showing 1 to 1 of 1 Previous | Next

← → Home Workspaces API Network

Search Postman

Invite ⚙️ 🔔 Upgrade

My Workspace New Import

Collections + ☰ MedRec Refactored ★

- POST Search patients
- POST Search patient winner
- POST Find new patients

Environments

History

MedRec Refactored / Search patients

POST http://12.0.0.10:10001/PatientSearch/jaxWsPatientFacadeBroker/fuzzyFindApprovedPatientsByLastnameAndSsn

Params Authorization Headers (10) Body Pre-request Script Tests Settings

Body (none) form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {"lastName": "", "ssn": ""}
```

Cookies Beautify

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 115 ms Size: 697 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {"retVal": [{"dob": "1975-04-26T00:00:00Z", "id": 2, "name": {"firstName": "Fred", "lastName": "Winner", "middleName": "I"}, "ssn": "123456789"}, {"dob": "1981-09-17T00:00:00Z", "id": 5, "name": {"firstName": "Gabrielle", "lastName": "Spiker", "middleName": "H"}, "ssn": "333333333"}, {"dob": "1969-03-13T00:00:00Z", "id": 3, "name": {"firstName": "Larry", "lastName": "Parrot", "middleName": "J"}, "ssn": "777777777"}, {"dob": "1982-03-18T00:00:00Z", "id": 6, "name": {"firstName": "Page", "lastName": "Trout", "middleName": "A"}, "ssn": "888888888"}]}
```

Online C

Find and replace Console

Postbot Runner Capture requests Cookies Trash

My Workspace New Import

Collections + ☰

Environments MedRec Refactored

History OMS Services OMS WebMVC

Overview POST Search patients POST Search patient winner POST Find new patients OMS Services +

MedRec Refactored / Search patient winner

POST http://12.0.0.10:10001/PatientSearch/jaxWsPatientFacadeBroker/fuzzyFindApprovedPatientsByLastnameAndSsn

Params Authorization Headers (10) Body Pre-request Script Tests Settings

Body none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {"lastName": "Winner", "ssn": ""}
```

Send Cookies Beautify

Status: 200 OK Time: 26 ms Size: 329 B Save as example

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON  

```
1 {
2   "RetVal": [
3     {
4       "dob": "1975-04-26T00:00:00Z",
5       "id": 2,
6       "name": {
7         "firstName": "Fred",
8         "lastName": "Winner",
9         "middleName": "I"
10      },
11      "ssn": "123456789"
12    }
13  ]
14 }
```

My Workspace

New Import

Overview POST Search patients POST Search patient winner POST Find new patients OMS Services +

Collections + ☰ MedRec Refactored ★

- POST Search patients
- POST Search patient winner
- POST Find new patients

Environments

History

OMS Services ★

OMS WebMVC ★

MedRec Refactored / Find new patients

POST http://12.0.0.10:10002/RegisterNewPatients/viewingNewlyRegisteredPatientsController/viewNewlyRegisteredPatients

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

Headers Key Value Description

<input checked="" type="checkbox"/> Content-Type	application/json	Description
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 132 ms Size: 922 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "RetVal": [  
3     {  
4       "id": 4,  
5       "version": 1,  
6       "email": "charlie@star.com",  
7       "password": "weblogic",  
8       "username": "charlie@star.com",  
9       "name": {  
10          "firstName": "Charlie",  
11          "lastName": "Florida",  
12          "middleName": "E"  
13        },  
14        "phone": "4151234564",  
15        "address": {  
16          "city": "Ponte Verde",  
17          "country": "United States",  
18          "state": "Florida",  
19          "street1": "235 Montgomery St",  
20          "street2": "Suite 15",  
21          "zip": "32301"  
22        },  
23        "dob": "1983-11-29T00:00:00Z",  
24        "gender": "MALE",  
25        "ssn": "4444444444",  
26      }  
27    ]  
28  }  
29}
```

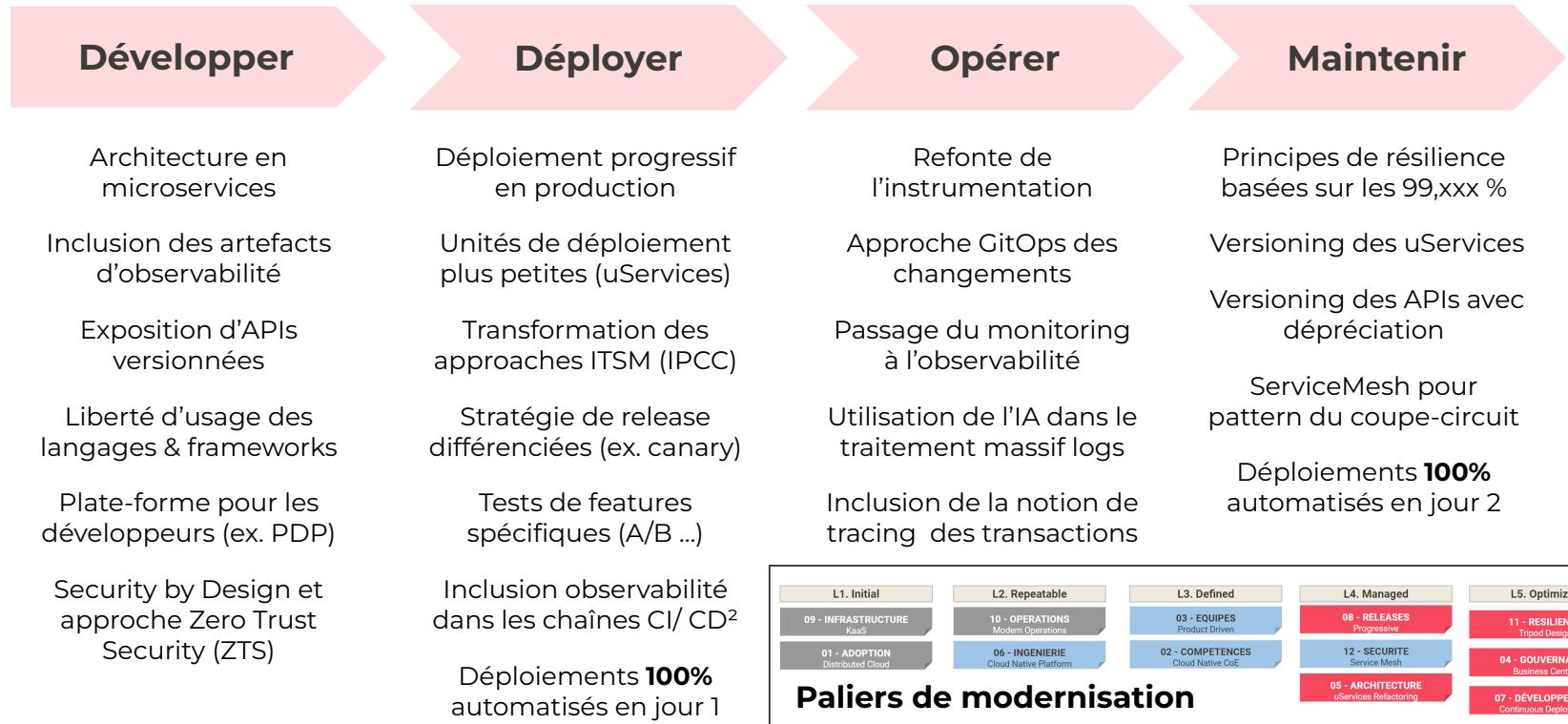
Postbot Runner Capture requests Cookies Trash

Cloud Native Innotation Labs

Step 9

OpenMind, expliquer comment cela aide à simplifier la modernisation, les nouvelles features, le cycle de release, les opérations ...

Cette modernisation impacte de multiples domaines





Thank you.

