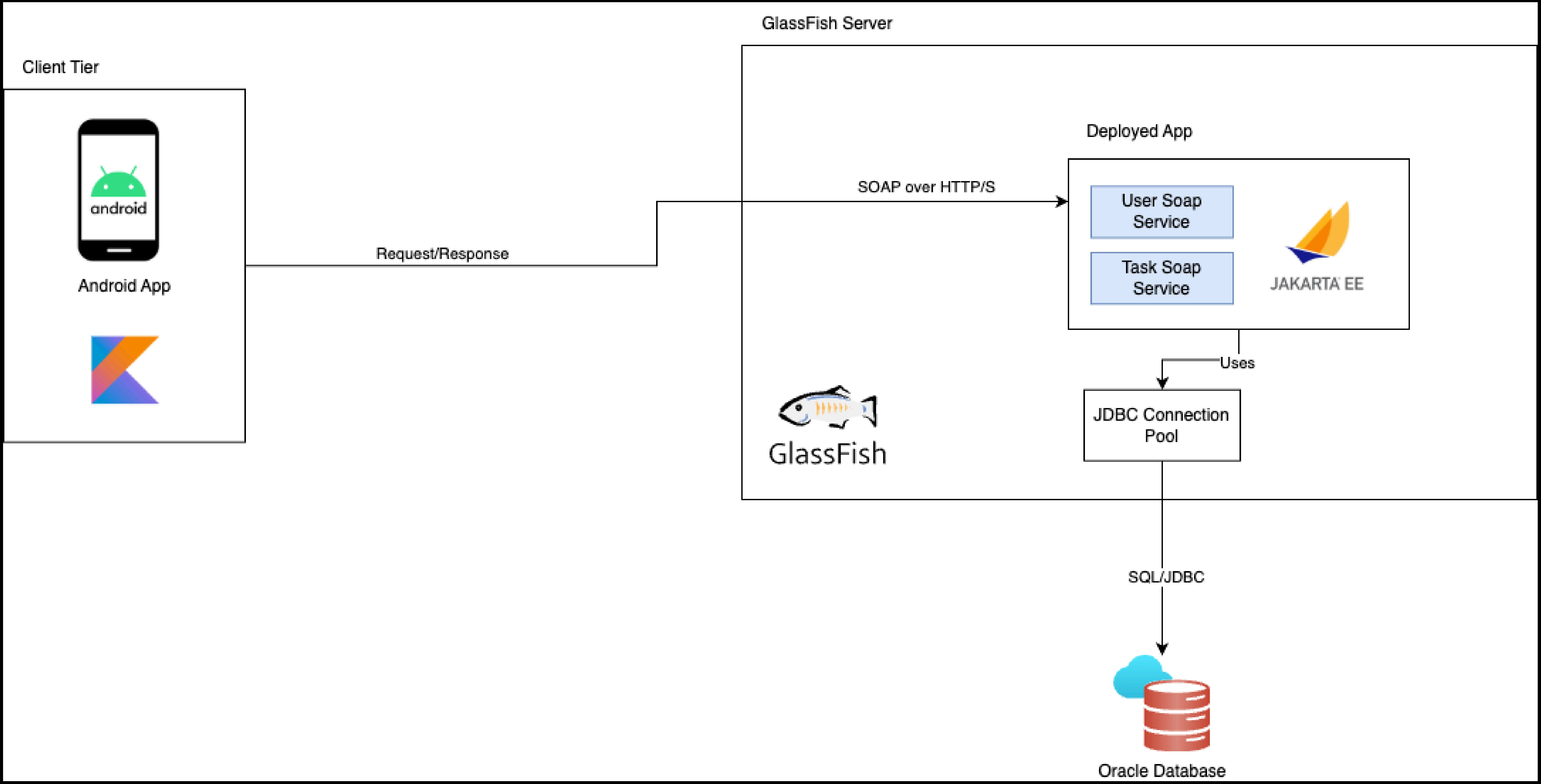
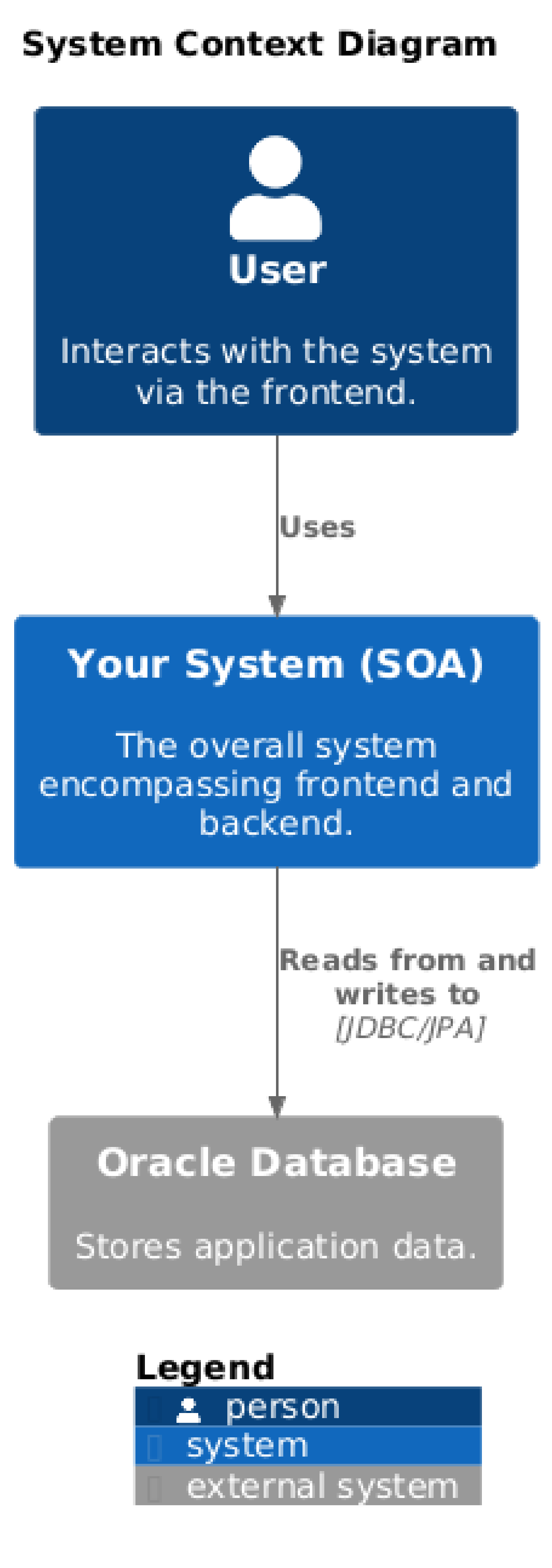
Architecture Framework

Nicolas Ceron

High Level Architecture



C4



System Context Diagram

User

Interacts with the system via the frontend.

Uses

Your System (SOA)

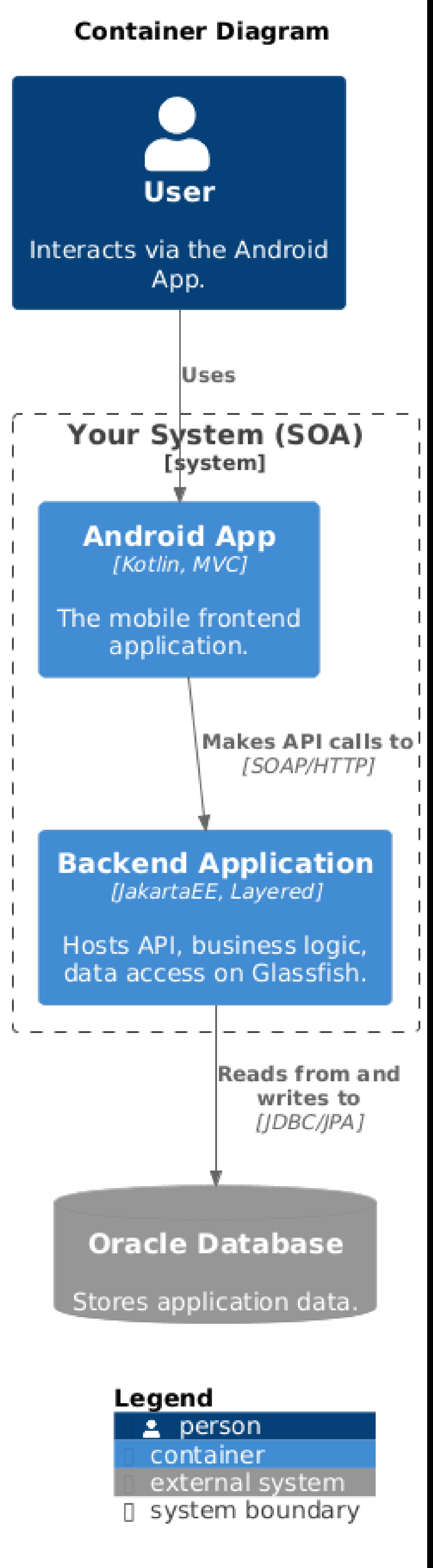
The overall system encompassing frontend and backend.

Reads from and writes to
[JDBC/JPA]

Oracle Database

Stores application data.

Legend
- person
- system
- external system



Container Diagram

User

Interacts via the Android App.

Uses

Your System (SOA)
[system]

Android App
[Kotlin, MVC]

The mobile frontend application.

Makes API calls to
[SOAP/HTTP]

Backend Application
[JakartaEE, Layered]

Hosts API, business logic, data access on Glassfish.

Reads from and writes to
[JDBC/JPA]

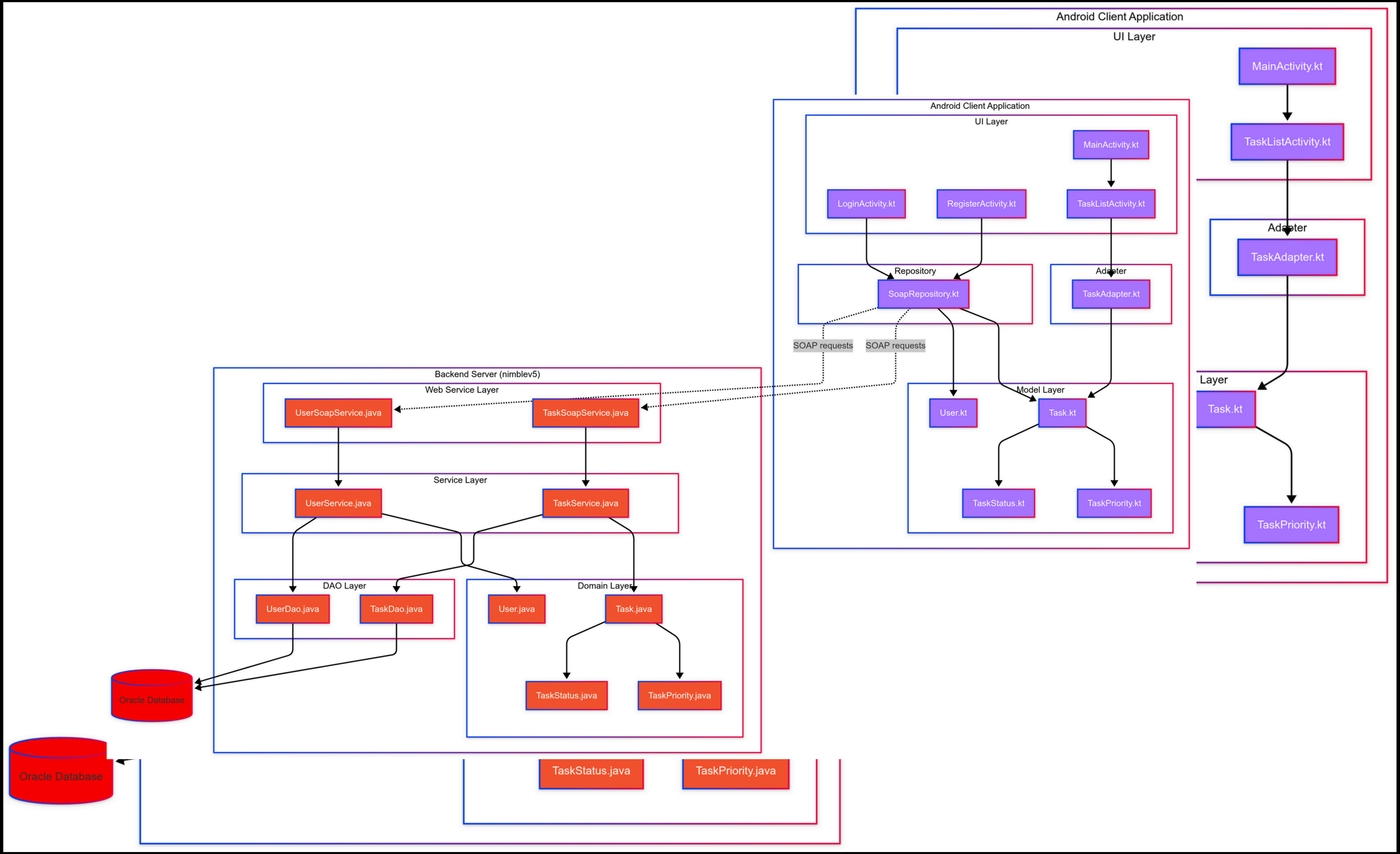
Oracle Database

Stores application data.

Legend
- person
- container
- external system
- system boundary

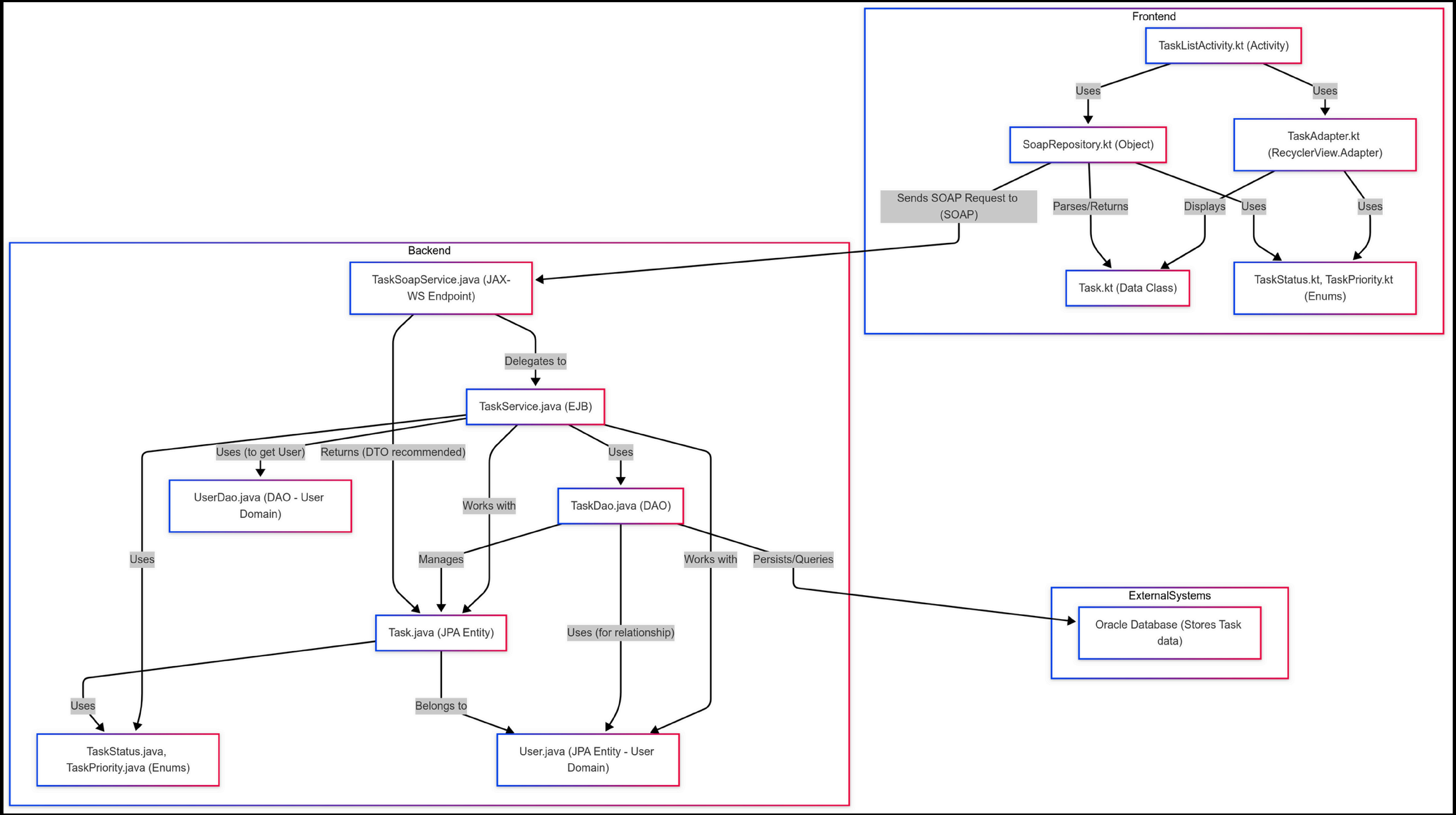
C4

Components



C4

Code Task



Frontend

TaskListActivity.kt (Activity)

Uses

Uses

SoapRepository.kt (Object)

TaskAdapter.kt (RecyclerView.Adapter)

Sends SOAP Request to (SOAP)

Parses/Returns

Displays

Uses

Uses

Task.kt (Data Class)

TaskStatus.kt, TaskPriority.kt (Enums)

Backend

TaskSoapService.java (JAX-WS Endpoint)

Delegates to

TaskService.java (EJB)

Uses (to get User)

Returns (DTO recommended)

Uses

UserDao.java (DAO - User Domain)

Works with

TaskDao.java (DAO)

Uses

Manages

Works with

Persists/Queries

Task.java (JPA Entity)

Uses (for relationship)

Uses

Belongs to

TaskStatus.java, TaskPriority.java (Enums)

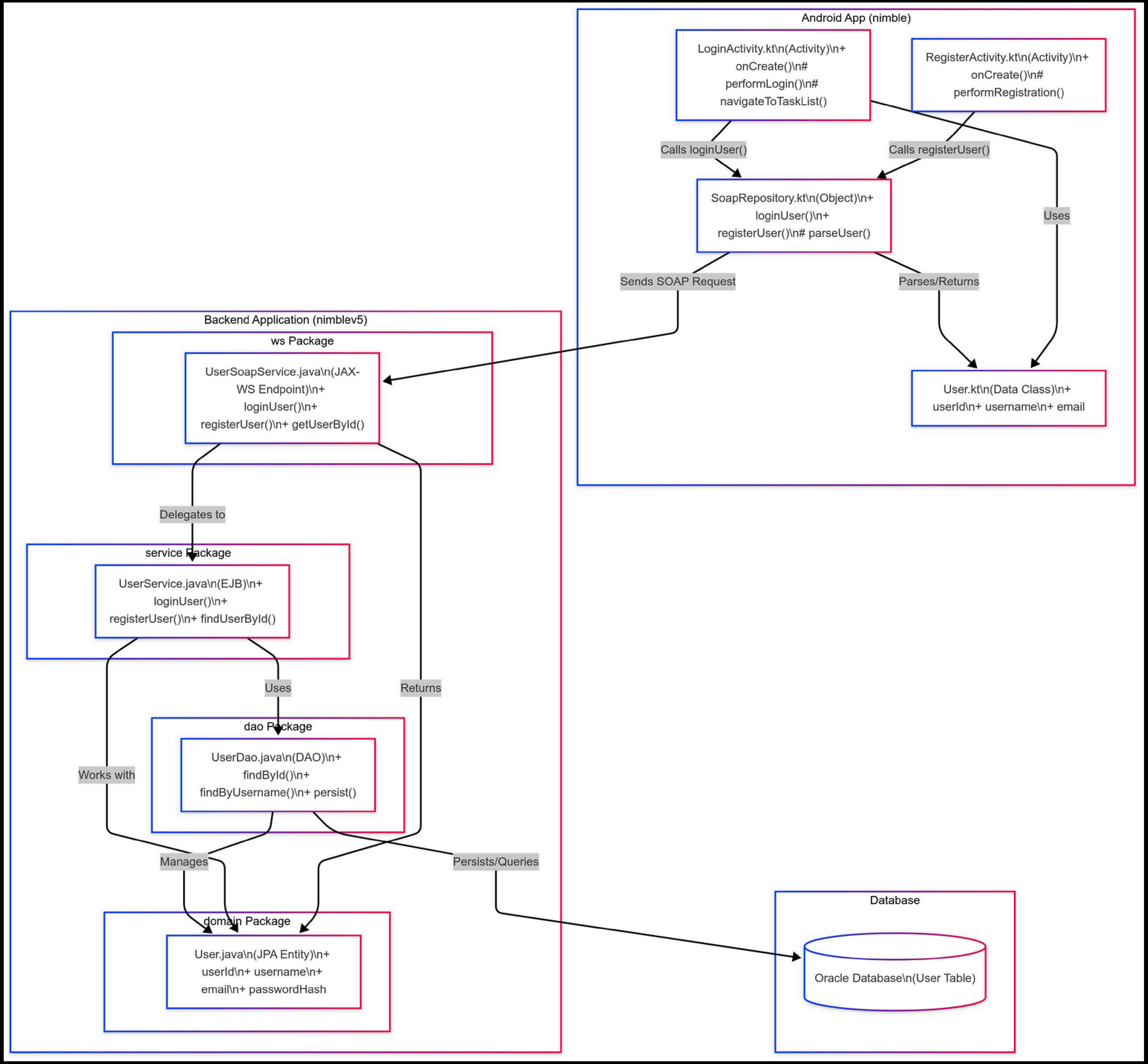
User.java (JPA Entity - User Domain)

ExternalSystems

Oracle Database (Stores Task data)

C4

Code User



Android App (nimble)

LoginActivity.kt\n(Activity)\n+ onCreate()\n# performLogin()\n# navigateToTaskList()

RegisterActivity.kt\n(Activity)\n+ onCreate()\n# performRegistration()

Calls loginUser()

Calls registerUser()

Uses

SoapRepository.kt\n(Object)\n+ loginUser()\n+ registerUser()\n# parseUser()

Sends SOAP Request

Parses/Returns

User.kt\n(Data Class)\n+ userId\n+ username\n+ email

Backend Application (nimblev5)

ws Package

UserSoapService.java\n(JAX-WS Endpoint)\n+ loginUser()\n+ registerUser()\n+ getUserById()

Delegates to

service Package

UserService.java\n(EJB)\n+ loginUser()\n+ registerUser()\n+ findUserById()

Uses

Returns

Works with

dao Package

UserDao.java\n(DAO)\n+ findById()\n+ findByUsername()\n+ persist()

Manages

Persists/Queries

domain Package

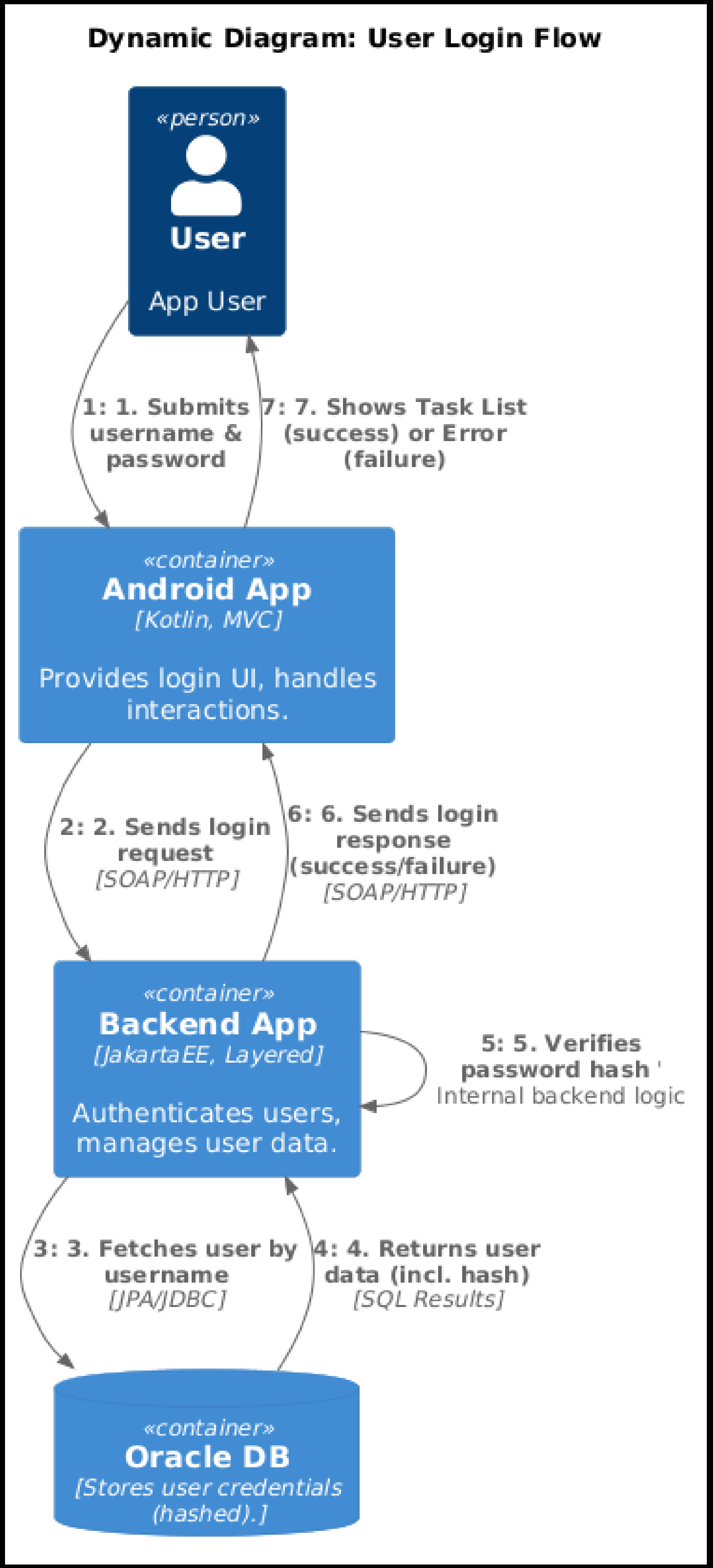
User.java\n(JPA Entity)\n+ userId\n+ username\n+ email\n+ passwordHash

Database

Oracle Database\n(User Table)

C4

Dynamic



Dynamic Diagram: User Login Flow

«person»
User
App User

1: 1. Submits username & password

7: 7. Shows Task List (success) or Error (failure)

«container»
Android App
[Kotlin, MVC]
Provides login UI, handles interactions.

2: 2. Sends login request *[SOAP/HTTP]*

6: 6. Sends login response (success/failure) *[SOAP/HTTP]*

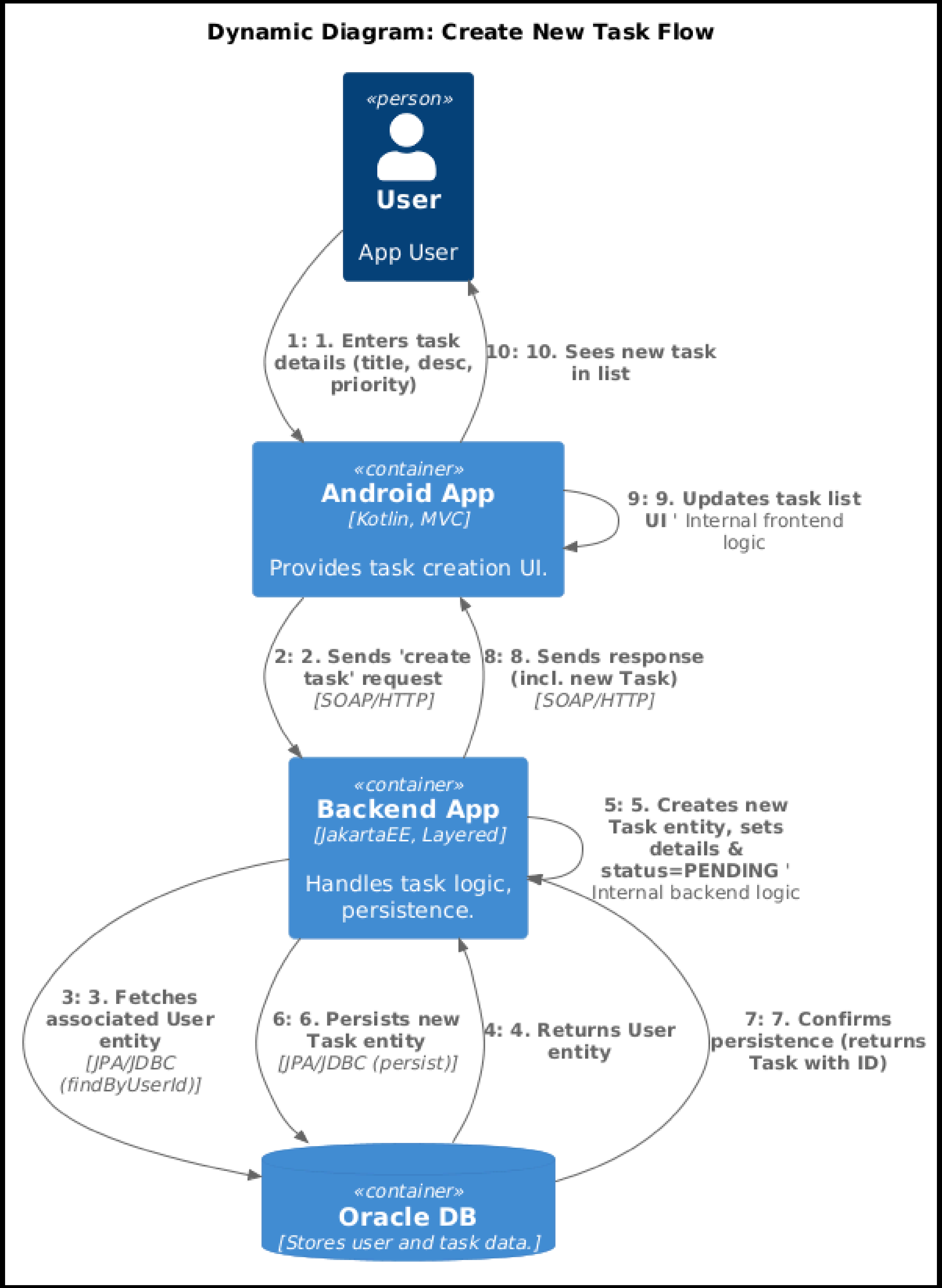
«container»
Backend App
[JakartaEE, Layered]
Authenticates users, manages user data.

5: 5. Verifies password hash ' Internal backend logic

3: 3. Fetches user by username *[JPA/JDBC]*

4: 4. Returns user data (incl. hash) *[SQL Results]*

«container»
Oracle DB
[Stores user credentials (hashed).]



Dynamic Diagram: Create New Task Flow

«person»
User
App User

1: 1. Enters task details (title, desc, priority)

10: 10. Sees new task in list

«container»
Android App
[Kotlin, MVC]
Provides task creation UI.

9: 9. Updates task list UI ' Internal frontend logic

2: 2. Sends 'create task' request *[SOAP/HTTP]*

8: 8. Sends response (incl. new Task) *[SOAP/HTTP]*

«container»
Backend App
[JakartaEE, Layered]
Handles task logic, persistence.

5: 5. Creates new Task entity, sets details & status=PENDING ' Internal backend logic

3: 3. Fetches associated User entity *[JPA/JDBC (findByUserId)]*

6: 6. Persists new Task entity *[JPA/JDBC (persist)]*

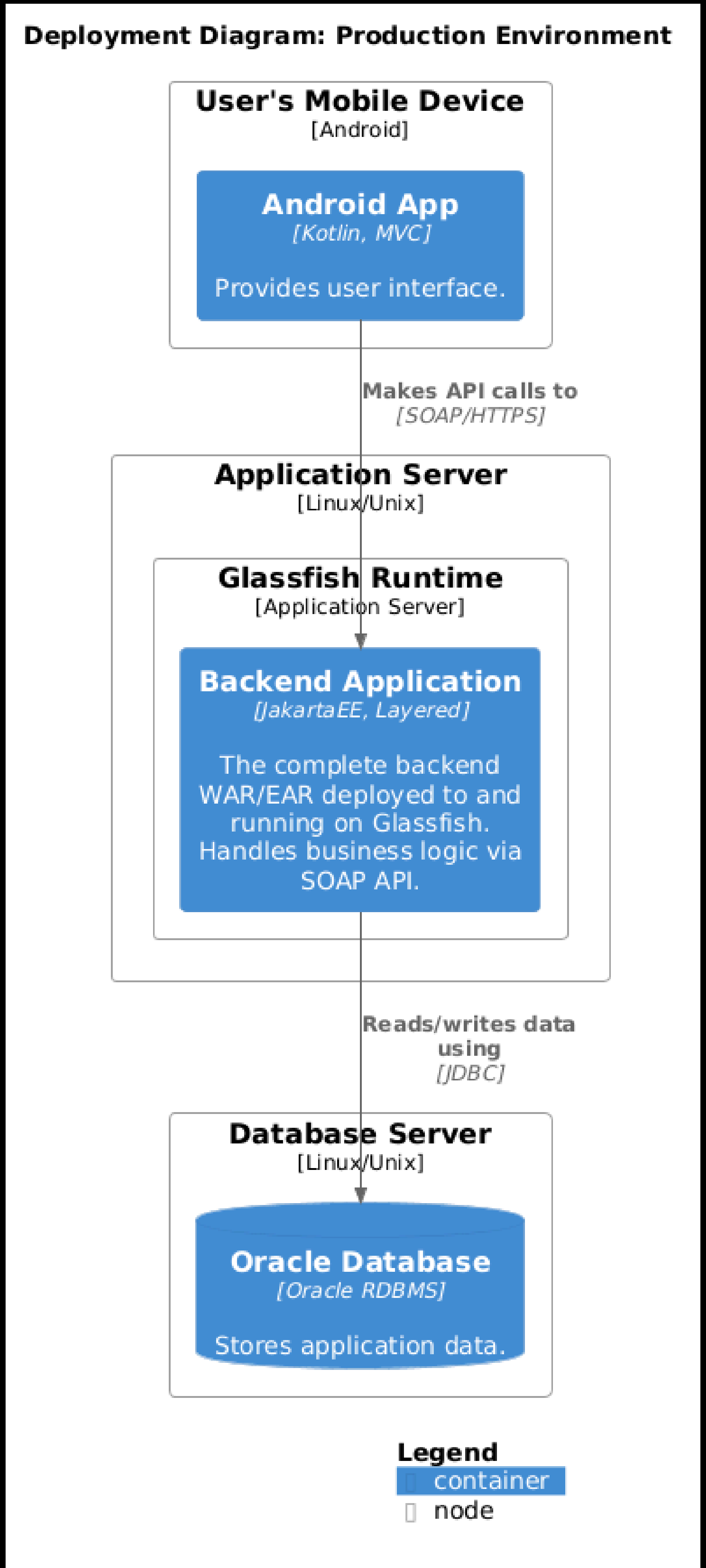
4: 4. Returns User entity

7: 7. Confirms persistence (returns Task with ID)

«container»
Oracle DB
[Stores user and task data.]

C4

Deployment



Deployment Diagram: Production Environment

User's Mobile Device
[Android]

Android App
[Kotlin, MVC]
Provides user interface.

Makes API calls to
[SOAP/HTTPS]

Application Server
[Linux/Unix]

Glassfish Runtime
[Application Server]

Backend Application
[JakartaEE, Layered]
The complete backend WAR/EAR deployed to and running on Glassfish. Handles business logic via SOAP API.

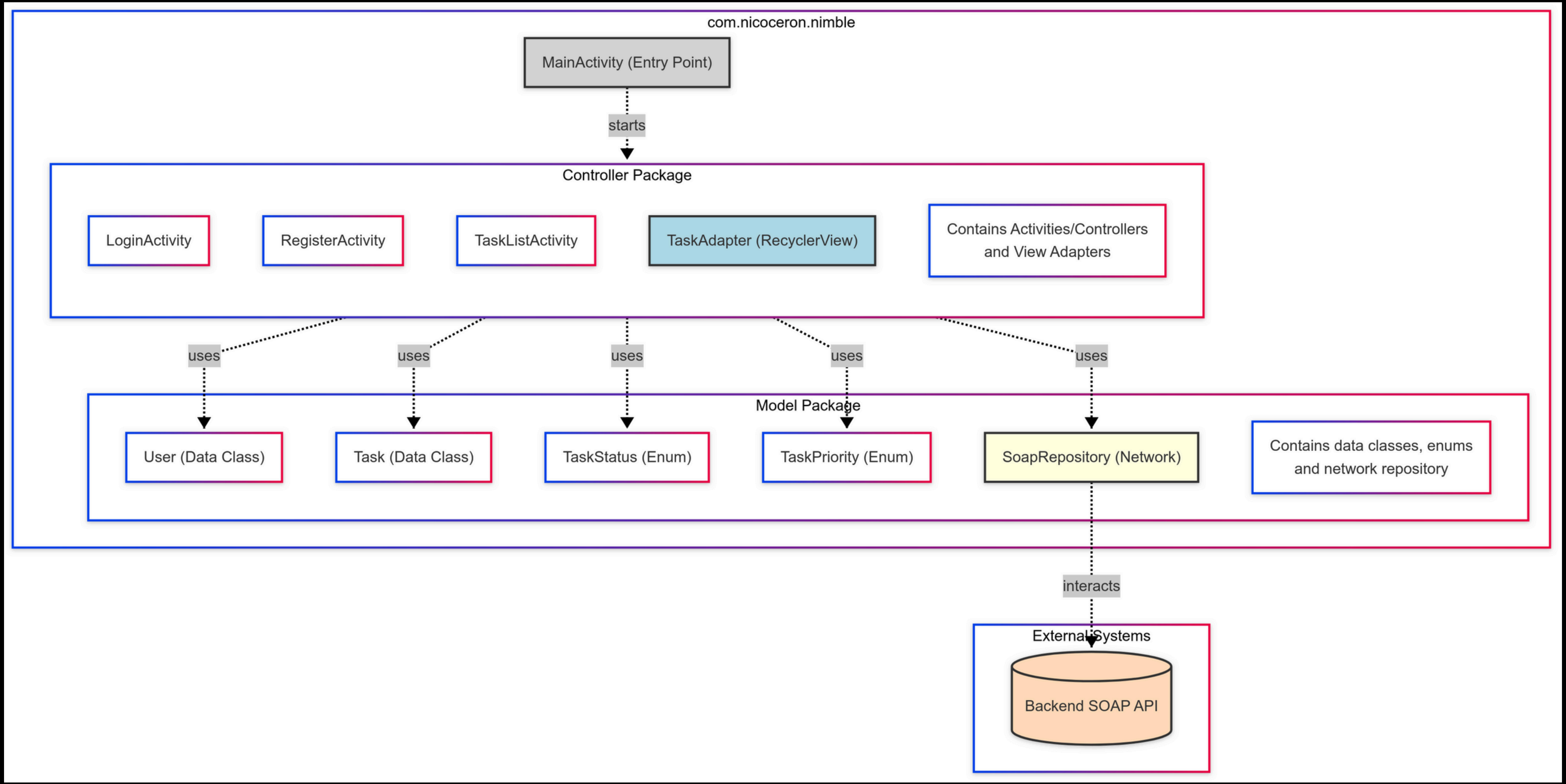
Reads/writes data using
[JDBC]

Database Server
[Linux/Unix]

Oracle Database
[Oracle RDBMS]
Stores application data.

Legend
- container
- node

Package Diagram (Front)



com.nicoceron.nimble

MainActivity (Entry Point)

starts

Controller Package

LoginActivity

RegisterActivity

TaskListActivity

TaskAdapter (RecyclerView)

Contains Activities/Controllers and View Adapters

uses

uses

uses

uses

uses

Model Package

User (Data Class)

Task (Data Class)

TaskStatus (Enum)

TaskPriority (Enum)

SoapRepository (Network)

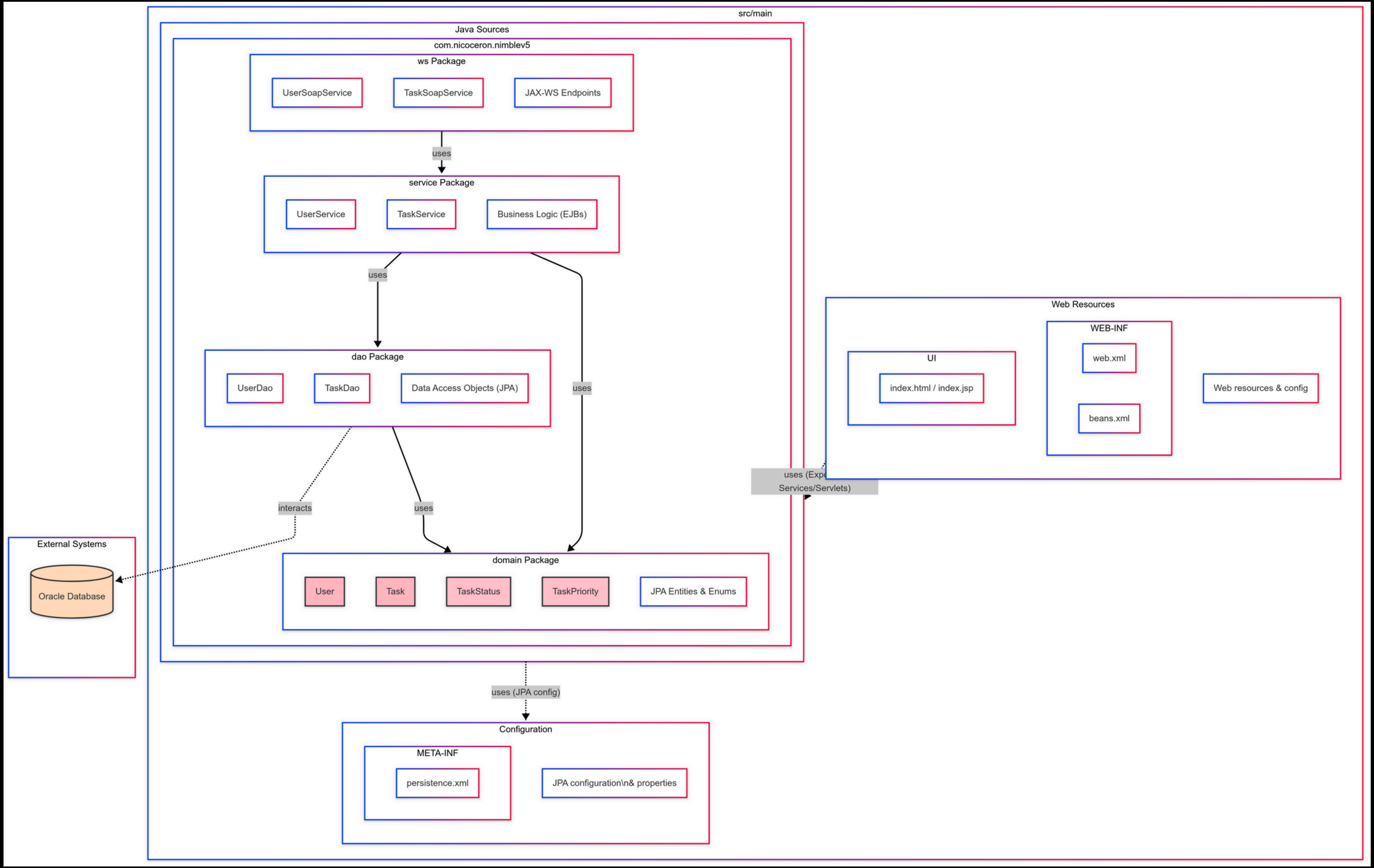
Contains data classes, enums and network repository

interacts

External Systems

Backend SOAP API

Package Diagram (Back)



SOLID Principles Analysis Matrix

Component/Pattern	SRP (Single Resp.)	OCP (Open/Closed)	LSP (Liskov Sub.)	ISP (Interface Seg.)	DIP (Dependency Inv.)
SOA (Principles)	Supports (Service=Capability)	Supports (Extend via services)	Contextual (Contract adherence)	Supports (Contract=Interface, maybe large)	Supports (Depends on contracts)
Android (Kotlin)	Facilitated (Kotlin features)	Facilitated (Kotlin extensions, IFs)	Facilitated (Kotlin null safety)	Facilitated (Kotlin interfaces)	Facilitated (Kotlin IFs, DI)
MVC (on Android)	Challenged (Controller overload)	Challenged (Controller modification)	Contextual	Challenged (Large implicit IFs)	Challenged (Direct C->V, C->M)
Jakarta EE (Platform)	Promotes (Components, CDI)	Promotes (Extensibility, Interceptors)	Supports (Std. interfaces, e.g., JPA)	Supports (Specific APIs)	Strongly Supports (CDI)
Java (Language)	Enables (OO, access mods)	Enables (IFs, Abstract Classes)	Enables (Inheritance, IFs)	Enables (Interfaces)	Enables (IFs, DI frameworks)
Layered Architecture	Supports (Layer=Responsibility)	Contextual (Extension vs Mod.)	Contextual (Depends on IFs)	Contextual (IFs between layers maybe large)	Challenged (Traditional Top-Down)
SOAP (Protocol)	N/A (Service design issue)	Supports (Extend via Headers/WS-*)	N/A (Service design issue)	N/A (Service design issue, WSDL=Interface)	N/A (Service design issue, WSDL=Abstract.)
Glassfish (Runtime)	N/A (Provides env.)	N/A (Provides env.)	N/A (Provides env.)	N/A (Provides env.)	N/A (Provides env. via CDI)
Oracle DB (System)	N/A (Applies to Stored Procs/Interact.)	N/A	N/A	N/A	N/A

Quality Attributes Impact Matrix

Component	Performance	Security	Maintainability	Scalability	Reliability/Avail.	Testability	Interoperability	Usability (Dev/End)
SOA (Principles)	Negative	Mixed/Contextual	Positive	Mixed/Contextual	Mixed/Contextual	Mixed/Contextual	Very Positive	Mixed (Dev) / NA (End)
Android OS	Positive	Positive (Platform)	Mixed (Framework)	Positive (Platform)	Positive (Platform)	Mixed (Framework)	Positive (Ecosystem)	Very Pos (End) / Pos (Dev)
Kotlin (Language)	Positive	Positive (Null safety)	Very Positive	Positive (Coroutines)	Positive	Positive	Very Pos (w/ Java)	Very Pos (Dev) / NA (End)
MVC (on Android)	Mixed/Contextual	Neutral	Very Negative (Complex)	Very Negative (Complex)	Negative	Very Negative	Neutral	Pos (Initial)/Neg (Dev)
Jakarta EE (Plat.)	Mixed/Contextual	Very Positive	Positive	Positive	Positive	Positive	Very Positive	Positive (Dev) / NA (End)
Java (Language)	Positive	Positive	Positive	Positive	Positive	Positive	Very Positive	Positive (Dev) / NA (End)
Layered Arch.	Negative	Positive	Positive (Intra-layer)	Mixed (Monolith)	Mixed/Contextual	Positive	Neutral	Positive (Dev) / NA (End)
SOAP (Protocol)	Very Negative	Positive (with WS-Sec)	Mixed/Contextual	Mixed/Contextual	Positive (with WS-Rel)	Mixed/Contextual	Very Positive	Negative (Dev) / NA (End)
Glassfish (Runtime)	Mixed/Contextual	Positive	Mixed/Contextual	Positive (Cluster)	Positive (Cluster)	N/A	Very Positive	Positive (Dev) / NA (End)
Oracle DB (System)	Very Positive	Very Positive	Very Negative (Complex)	Very Positive (RAC/Shard)	Very Positive (RAC/DG)	N/A	Very Positive	Negative (Dev) / NA (End)

Architectural Tactics Support

Component	Availability (Redundancy, Failover, Isolate)	Performance (Cache, Pool, Concurrency)	Modifiability (Encaps., Interfaces, DI)	Security (AuthN/Z, Encrypt)	Testability (Mocking, Isolate)
SOA (Principles)	Supports (Isolate), Contextual (Redund./Failover)	Hinders (Latency), Supports (Async)	Supports (Interfaces, Encaps.)	Supports (Per-service)	Contextual (Isolate vs Int.)
Android OS	Managed by OS	Supports (ART, Threads)	Framework (Components)	Supports (Perms, Sandbox)	Hinders (Framework Dep.)
Kotlin (Language)	Supports (Coroutines)	Supports (Coroutines, Inline)	Supports (OO, Functional)	Supports (Null Safety)	Supports (Testing Frameworks)
MVC (on Android)	Hinders	Hinders	Hinders (Coupling)	Hinders	Hinders (Controller)
Jakarta EE (Plat.)	Supports (Cluster, Tx)	Supports (Pool, JPA Cache, Concurr.)	Strongly Supports (CDI, Interfaces)	Strongly Supports (Security)	Supports (CDI Mocks)
Java (Language)	Supports (Threads, Exceptions)	Supports (Threads, JIT, GC)	Supports (OO, Interfaces)	Supports (Security Mgr)	Supports (JUnit, Mocking)
Layered Arch.	Supports (Isolation)	Hinders (Latency)	Supports (Separation of Concerns)	Supports (Layer Boundaries)	Supports (Layer Isolation)
SOAP (Protocol)	Supports (WS-Rel, Faults)	Hinders (XML), Supports (MTOM)	Supports (WSDL Contract, Extend)	Strongly Supports (WS-Sec)	Contextual (Requires Tools)
Glassfish (Runtime)	Strongly Supports (Cluster, Failover)	Supports (Pools, Tuning)	Supports (Modularity, Hot Deploy)	Supports (Jakarta EE Sec.)	Facilitates (Test Env.)
Oracle DB (System)	Strongly Supports (RAC, DG, RMAN)	Strongly Supports (Cache, Index, IM)	Supports (PL/SQL, Part.)	Strongly Supports (TDE, RBAC)	N/A

Patterns Matrix

Component	Microservices	EDA	API Gateway / ESB	Observer	Factory/ Singleton	Repository / DAO	DI / IoC	ORM (JPA)	Others
SOA (Principles)	Precursor	Complementary	ESB (Common)	Possible	N/A	N/A	Via Contracts	N/A	Service Contract, Discovery
Android (Kotlin/MVC)	N/A (Client)	Possible (EventBus)	N/A	Key (UI)	Common	Common (Repo)	Key (Hilt)	Room (ORM-like)	MVP, MVVM, MVI (MVC Alts)
Jakarta EE (Java/Layered)	Enables	Supports (JMS/MDB)	Supports (JAX-RS)	CDI Events	CDI (Managed)	Common (DAO/Repo)	Fundamental (CDI)	Key (JPA)	EJB Patterns, Front Ctlr
SOAP (Protocol)	Used in (rare)	Via JMS	Used with	N/A	N/A	N/A	N/A	N/A	RPC, WSDL
Glassfish (Runtime)	Deploys	Deploys (MDB)	Deploys	N/A	N/A	N/A	Provides (CDI)	Provides (JPA)	Implements Jakarta EE
Oracle DB (System)	Backend for	Backend for	Backend for	N/A	N/A	Accessed by	N/A	Accessed by	RAC (Cluster), DG (Standby)

Job Market Demand Matrix

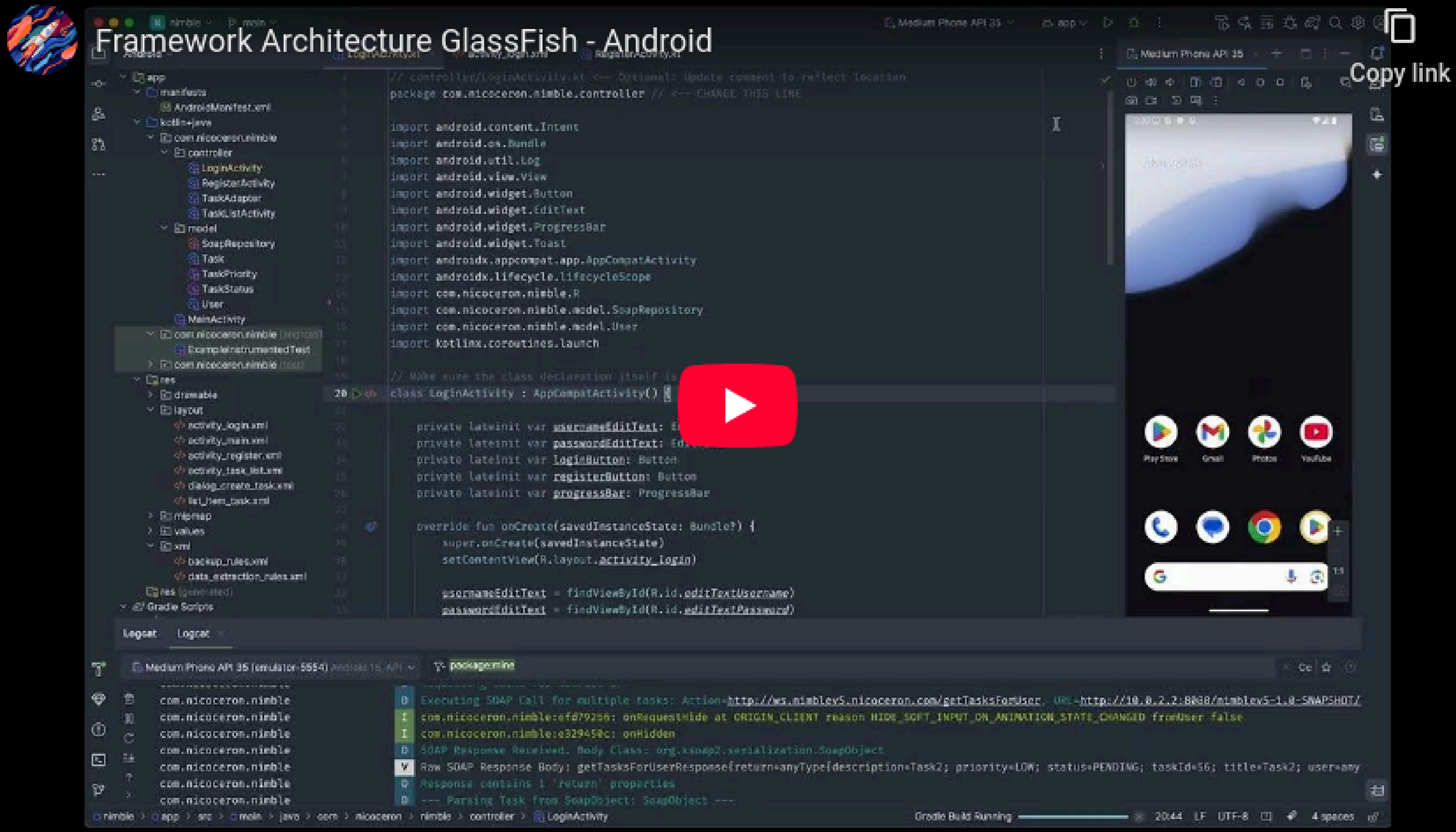
Skill/Component	General Demand Trend	Bogotá/CO Demand (Level)	Key Skills Mentioned	Typical Experience	Indicative Salary (Bogotá)
SOA (Concepts)	Stable (Implicit)	Medium (Implicit)	Service Design, Integration, API (SOAP/REST), Microservices	Senior/Architect	High
Android Dev	Very High	High	Kotlin, Android SDK, Jetpack (Compose, MVVM), REST, Git	All levels	Variable (High for Senior/Remote USD)
Kotlin	Very High (Growing)	High	Kotlin Lang, Coroutines, Java Interop, Functional	All levels	Variable (Tied to Role)
MVC (Android Pattern)	Low (Replaced)	Low	(Usually MVP/MVVM/MVI requested instead)	N/A	N/A
Jakarta EE	Stable/Growing	Medium/High (Implicit)	APIs (JPA, REST, CDI, JMS), App Servers (WildFly, etc.)	Mid/Senior	Variable (Tied to Java role)
Java	Very High (Stable)	High	Core Java, Spring Boot, Jakarta EE APIs, SQL, Cloud	All levels	Variable (High for Senior)
SOAP	Decreasing (New Proj.)	Low/Medium (Niche)	XML, WSDL, WS-Security, JAX-WS, Legacy Integration	Mid/Senior	Variable (Tied to Integration role)
Glassfish Admin/Dev	Very Low / Decreasing	Very Low / Non-existent	(General Jakarta EE Admin Skills)	N/A	N/A
Oracle DBA/Dev	Stable (Evolving)	High	Oracle Admin, SQL, PL/SQL, Tuning, RAC, DG, OCI	Senior	High (e.g., 12M+ COP/month Senior)

Strengths:

- Robust Backend: Jakarta EE/Java + Oracle DB = Proven, reliable, secure, scalable for enterprise needs.
- Leading Mobile: Android + Kotlin = Top modern platform, vast market, active ecosystem.
- Relevant SOA Principles: Core concepts (reusability, loose coupling) still valuable, influenced microservices.
- Interoperability: SOAP/SOA historically strong for connecting diverse systems.

Weaknesses:

- Legacy Tech: SOAP & Glassfish declining vs. REST/modern servers. Risks: Less support/talent, obsolescence.
- Android MVC Pattern: Suboptimal vs. MVVM/MVP/MVI; impacts maintainability/testability.
- Complexity & Cost: High admin/licensing (Oracle DB), management/governance (SOA/ESBs).
- SOAP Performance: XML-based, generally less performant than alternatives (e.g., REST).



```
// controller/LoginActivity.kt <-- Optional: Update comment to reflect location
package com.nicoceron.nimble.controller // <-- CHANGE THIS LINE

import android.content.Intent
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.ProgressBar
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.lifecycle.lifecycleScope
import com.nicoceron.nimble.R
import com.nicoceron.nimble.model.SoapRepository
import com.nicoceron.nimble.model.User
import kotlinx.coroutines.launch

// Make sure the class declaration itself is
class LoginActivity : AppCompatActivity() {

    private lateinit var usernameEditText: E
    private lateinit var passwordEditText: Ed
    private lateinit var loginButton: Button
    private lateinit var registerButton: Button
    private lateinit var progressBar: ProgressBar

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        usernameEditText = findViewById(R.id.editTextUsername)
        passwordEditText = findViewById(R.id.editTextPassword)
```

Medium Phone API 35 (emulator-5554) Android 16, API

package:mine

```
com.nicoceron.nimble   D  Executing SOAP Call for multiple tasks: Action=http://ws.nimblev5.nicoceron.com/getTasksForUser, URL=http://10.0.2.2:8080/nimblev5-1.0-SNAPSHOT/
com.nicoceron.nimble   I  com.nicoceron.nimble:efd79256: onRequestHide at ORIGIN_CLIENT reason HIDE_SOFT_INPUT_ON_ANIMATION_STATE_CHANGED fromUser false
com.nicoceron.nimble   I  com.nicoceron.nimble:e329450c: onHidden
com.nicoceron.nimble   D  SOAP Response Received. Body Class: org.ksoap2.serialization.SoapObject
com.nicoceron.nimble   V  Raw SOAP Response Body: getTasksForUserResponse{return=anyType{description=Task2; priority=LOW; status=PENDING; taskId=56; title=Task2; user=any
com.nicoceron.nimble   D  Response contains 1 'return' properties
com.nicoceron.nimble   D  --- Parsing Task from SoapObject: SoapObject ---
```

nimble > app > src > main > java > com > nicoceron > nimble > controller > LoginActivity

Gradle Build Running 20:44 LF UTF-8 4 spaces