Architectural specification

Oyama Bilana u21815349

May 3, 2021

0 Introduction

0.1 Problem description

The purpose of this system is to serve a delivery company by making delivery and driver management more efficient. The business wants to be able to let customers track their deliveries easier and avoid logging queries. It will be important for the new system to capture arriving parcel information and aid with delivery scheduling. It is recommended that the system be developed such that drivers and customers can get parcel information on the go while office workers can capture and schedule deliveries from desktops in the office. Hence this system should be deployed on desktops and mobile.

0.2 Business benefits

The new system will be beneficial in Maintaining correct information about parcels, Setting optimal/efficient delivery schedules and routes hence cost saving, Customers will be happier because they will see an update on their shipment, drivers will be on time as they will always have access to their schedule wherever they and the summary and history reports will be more accurate and timely, leading to more insights and better reactions.

0.3 User Characteristics

User: Parcel clerk
Role : Captures incoming parcels information at the front desk
Education : Matric/Diploma
Experience : 0-1 years of office work experience
Expertise: Has expertise in Data entry, Customer support and report writing.
technical skills : Should be able to use Windows applications
System use : Search and view parcels, Generate reports and Capture new parcels into database

User: Delivery Clerk
Role : Handles delivery schedules and manages drivers
Education : Matric/Diploma
Experience : 0-1 years office work experience
Expertise: Data entry and Report writing

Technical skills : Using Windows applications
System use : Generate reports and Create delivery schedules

User: General manager
Role : Supervises all general operations, manages delivery schedules, reporting and drivers
Education : Diploma/Degree
Experience : 1 year managing experience
Expertise: Report writing, planning and data entry
Technical skills : Use Windows applications
System use : Generate reports, Manager drivers and scheduling

User: Delivery driver
Role : Delivers parcels to customers on a schedule
Education : Matric
Experience : Should be able to read and write
Technical skills : Should be able to use mobile phone
System use : View delivery schedule and connect with customer

User: Customer
Role : Requests parcel delivery status
Education : High school
Technical skills : Should be able to use mobile phone
System use : Track parcel delivery

1 Non-Functional requirements

1.1 Usability

Source of Stimulus - End user
Stimulus - Capture Parcel information
Environment - Run-time
Artifact - Parcel management subsystem
Response - User successfully capture parcel details
Response Measure - Successfully after first training session

Source of Stimulus - End users
Stimulus - Tries to Track parcel
Environment - Run-time
Artifact - Parcel management subsystem
Response - User gets desired output
Response Measure - Within 1 minute without training

Tactics - Provide Cancel and undo Maintain system model - Give user progress feedback

1.2 Interoperability

Source - Delivery scheduling subsystem
Stimulus - List of parcel names sent
Artifact - Parcel management subsystem
Environment - subsystems known prior to run-time
Response - Filter specified parcels and send to Delivery scheduling System
Response Measure - Information passed correctly 99% of the time

Source - Reporting subsystem
Stimulus - Time period information sent
Artifact - parcel management subsystem, Driver management subsystem, Delivery scheduling subsystem
Environment - subsystems known prior to run-time
Response - Filter information for given period, send to reporting subsystem.
Response Measure - Information passed correctly 99% of the time.

Tactics - Orchestrate between services by using a mediator

1.3 Availability

Source of stimulus - Ping test
Stimulus - Incorrect Response
Artifact - Processor, Communication channel, db Storage
Environment - Normal operation
Response - Report to support team, Log fault, make temporarily unavailable
Response measure - Error should be less than 1%

Source of stimulus - Monitor
Stimulus - Server crashed
Artifact - Processor
Environment - Normal operation
Response - Report to support team
Response measure - Replacement should be completed with 2 hours

Tactics - Ping/echo Monitor

1.4 Maintainability

Source - Developer
Stimulus - Allocate more RAM and CPU to delivery scheduling subsystem
Artifact - resource
Environment - Run time
Response - Change made and component tested
Response measure - Disruptions caused by this modification should be rectified within 2 hours. Other subsystem must not need modification to suit this modification.

Source - Developer
Stimulus - Change GUI
Artifact - user interface, code
Environment - Design time
Response - Change made and Unit tested
Response measure - Must require 0% modification to other components
-A new developer should be able to make changes without knowledge of other components
-It must take less than 5 days to change and deploy

Tactics - Reduce coupling. Late binding of system resources

1.5 Performance

Source of Stimulus - End user
Stimulus - Initiate parcel sorting function
Artifact - Delivery scheduling subsystem
Environment - normal operation
Response - Function is processed
Response measure - Sort a batch of packages time less than 30 minutes. Calculate routes time less than 30 minutes

Source of stimulus - Delivery schedule subsystem
Stimulus - Request for parcel details
Artifact - parcel management system
Environment - normal operation
Response - Request is processed
Response measure - finish operation in 20 seconds

Tactics - Introduce Concurrency. Increase resources

1.6 Security

Source - Delivery driver Stimulus - Attempts to alter delivery schedule
Artifact - Driver management System, data
Environment - system is online
Response - Only provide access to authorized users. Log system activities
Response measure - Should withdstand 97% of unauthorized access

Source - Outsider attacker
Stimulus - Attempts DDOS attack
Artifact - Entire System
Environment - System is online and behind firewall
Response - Notify support team of increased requests. Only provide access to authorized users
Response mearsure - 9.8/10 attacks should be resisted
Tactics - Authorize actors. Detect service denial

2 Architectural design discussion

Challenge

The application is user driven and must serve the user goals with few errors and be as efficient as possible . Thefore it must be highly Usable. From a budget constraint it must be deployed on the cloud. The micro-services architecture with client dedicated API Gateways will suffice.

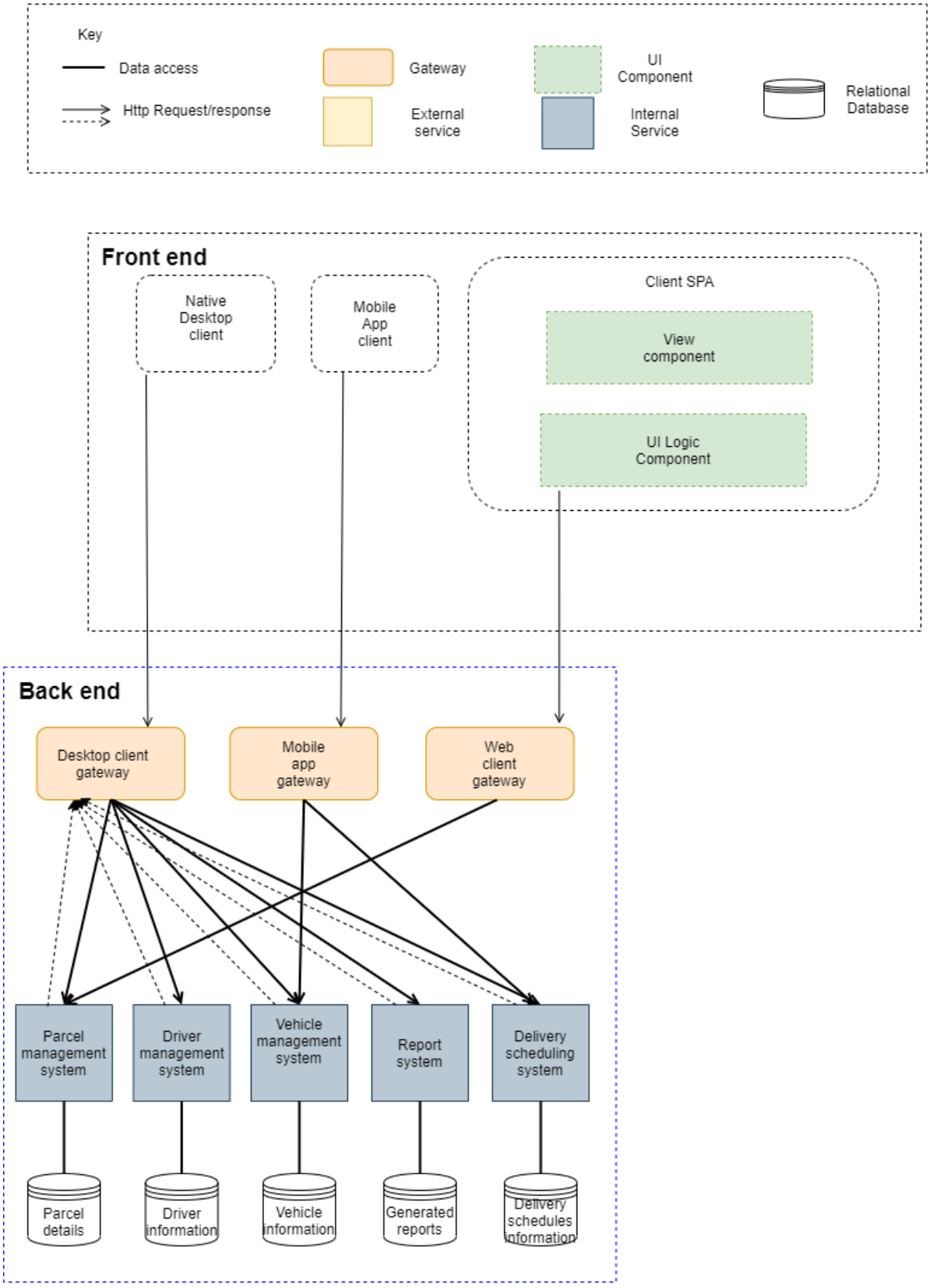
Why micro-services is relevant?

Promotes semantic coherence
restricted dependencies
defer binding
Freedom in Technology, thus most cloud providers will suffice.
Modifiability, thus UI can be developed and modified independently, allowing user feedback, thus enhancing usability
API Gateways ensures Authorization

Consequences of design decisions
Not having full control of physical hardware
Some performance trade-off by decoupling services

What factors influenced the decisions?
Budget
Main Quality requirement, which is Usability

2.1 Architectural patterns diagram



2.2 Architectural styles

Tactics for Augmenting Architectural pattern
Modifiability :

- Reduce coupling by allowing Inter-service comm through API Gateway

Security :

- Introduce API Gateway pattern to handle Authorization and routing

Availability :

- Back-end to front end pattern (BFF) to split API Gateway into multiple Gateways to serve each client type
- BFF reduces the probability of API Gateway being a single point of failure

Tactics to support quality attributes
NB: Some tactics are already mentioned in the Architectural pattern, but we define them again to ensure completeness.

Performance
Introduce Concurrency - Operations of the Delivery subsystem lend themselves to parallel processing
Increase resources - Add a faster processor and more memory

Availability
Ping/echo
Monitor

Usability
Provide Cancel and undo
Maintain system model - Give user progress feedback

Maintainability
Reduce coupling
Late binding of system resources

Interoperability
Orchestrate between services by using a mediator

Security
Authorize users
Detect service denial

2.3 Architectural constraints

Business/Organisational
Resources(budget) - Must use pay per use cloud provider because of low upfront budget
Legal factors - Ensure customer information privacy

Technical

Hardware
-Desktop app must run on minimum 2GHz processor, 4GB RAM
-Mobile App needs at least 1.5 GB RAM

Operating System
-Desktop app must support windows
-Mobile App must support Android v.6.0.1 upwards
-Web client must run on all major browsers

Programming constraints
–Must use Language with multi-threading support for back-end delivery schedule component

2.4 Actor-System Interaction

2.4.1

<table>
<tr><td colspan="2">Use case: Assign driver to vehicle</td></tr>
<tr><td colspan="2">Preconditions:
1. User must be logged in
2. Driver management subsystem must be available
3. Vehicle information must exist
4. Driver information must exist</td></tr>
<tr><td>Actor: Delivery clerk</td><td>System: Vehicle Management System</td></tr>
<tr><td>1.User clicks on “driver management”

3. Searched and selects driver and vehicle. Then clicks on “Assign driver to vehicle”</td><td>0.Display Home page window

2. Display driver management page with list of drivers and vehicles

4.System verifies errors and displays confirmation message</td></tr>
<tr><td colspan="2">Postcondition: System data is updated; driver is associated with a vehicle</td></tr>
<tr><td colspan="2">Exceptions:
If user is not logged in, don’t show homepage
If subsystem is not available, notify user with an error message
If vehicle info not found, notify user</td></tr>
</table>

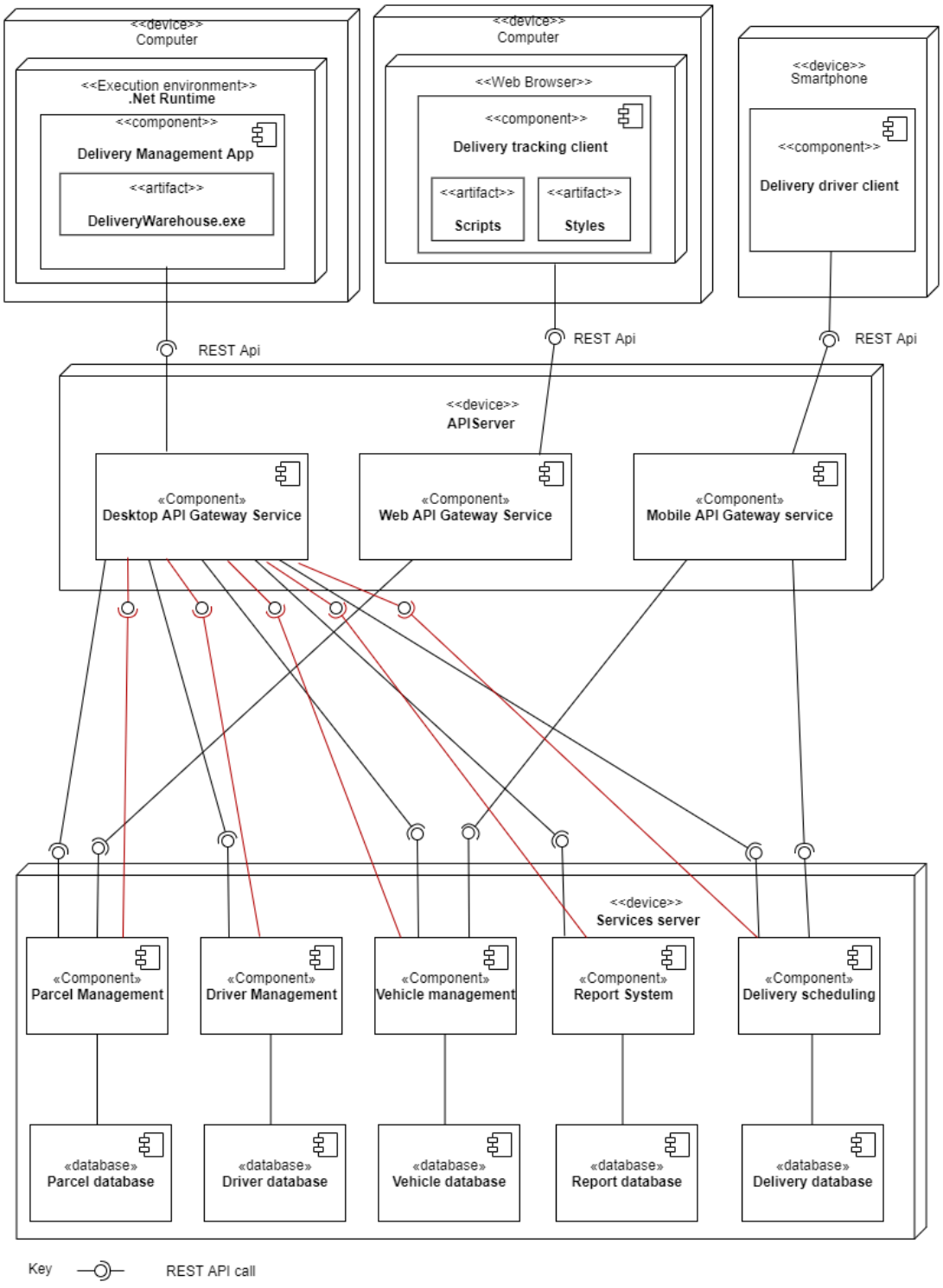
2.4.2

Use case: Add/Update Vehicle information	
Preconditions: 1. User must be logged in 2. Vehicle subsystem must be available 3. Vehicle information must exist	
Actor: Delivery clerk	System: Vehicle Management System
1.User clicks on “vehicle management” 3. Enters information and clicks on “Search” 5. User selects vehicle 7. User enters information and clicks “Update vehicle information”	0.Display Home page window 2. Display vehicle management page with list of vehicle information 4. Displays filtered results 6. Displays form with editable fields 8. Displays confirmation or error message
Postcondition: Updated driver details are available to be used	
Exceptions: If user is not logged in, don’t show homepage If subsystem is not available, notify user with an error message If vehicle info not found, notify user	

2.4.3

Use case: Track vehicle location	
Preconditions: 1. User must be logged in 2. Vehicle subsystem must be available 3. Vehicle information must exist	
Actor: Delivery clerk	System: Vehicle Management System
1. User clicks on “vehicle management” 3. Selects vehicle and clicks “Track vehicle” 5. Clicks “Done”	0.Display Home page window 2. Display vehicle management page 4. Display vehicle location or show error message 6. Closes vehicle information window
Postcondition: System state not changed	
Exceptions: If user is not logged in, don’t show homepage If subsystem is not available, notify user with an error message If vehicle info not found, notify user	

2.5 Deployment model



2.6 Technology requirements

Client Technologies
Desktop client
Windows Forms (Highly optimized for windows)
Intel core i5 7th Gen Desktop with 8GB RAM

Mobile Client
Java. Native on android and development team is experienced in java.

Web Client
Vanilla JavaScript and (HTML & CSS)

Client to back-end
Protocol - REST (Good support for browser client)

Back-end
API Gateway
Kong Gateway. It is open source and promotes high performance, extensibility and authentication

Genetic programming code
Use Java for computations. Java has good multi-threading support.
16 Core Computing Engine with 32GB RAM

Other subsystems
Use Java. Object Oriented and allows separation of concerns.
Promotes good code structure with high security.

Database
Use Relational database.
Offers relational data integrity
Promotes faster query responses