

SMART INVENTORY MANAGEMENT & OPTIMIZATION (SIMO)

Project S34 Review 2



SMART INVENTORY MANAGEMENT & OPTIMIZATION (SIMO)

Project S34 Review 2



OUR TEAM



Chester
Lim Zi Hao



Cui Wenqi



Olivia
Chen Ken



Justin
Peng Zheng
Wei



Chia Yu Ying



Hour
Youlin Sereydevid



Sun Kairan

TABLE OF CONTENTS

01

OVERVIEW

Company Background, Problem Framing, Problem Analysis, Milestones, User Needs & Constraints, Refined Problem Statement

02

DESIGN DIRECTION

Concept Generation, Optimizations Research & Analysis, Web Application Design

03

RISK ASSESSMENT

Consequence Frequency Matrix, Potential Risk & Precautions

04

PROJECT MANAGEMENT

Project Task Allocation, Project Schedule, Budget Allocation

05

CONCLUSION

Overview

Company Background

Problem Framing

Problem Analysis

Milestones

User Needs & Constraints

Refined Problem Statement

Company Background¹

- Singapore Airlines Engineering Company (SIAEC) is Asia's foremost maintenance, repair and overhaul (MRO) provider
- Delivers integrated solutions to large client bases consisting of both international airline and aerospace companies.
- Provides frontline maintenance services to more than 60 airlines* that fly through Singapore whilst ensuring a high level of punctuality for their customers' flight takeoffs.

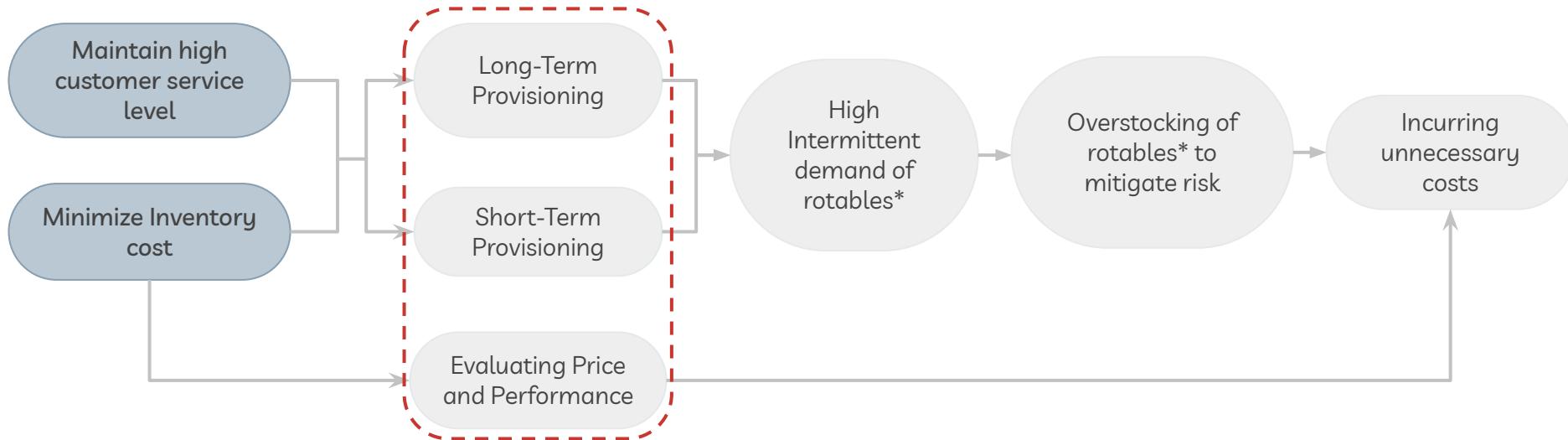
*Some companies include Singapore Airlines, SilkAir and TigerAir

Problem Framing

Despite efforts by SIAEC in deploying inventory management practices, the company ***still overprovision rotables***, incurring ***unnecessary capital***. The underlying causes of this overprovisioning include ***inefficient inventory management practices, forecasting techniques, and inflexible interface for managing inventory.***

Problem Analysis

Goals for SIAEC

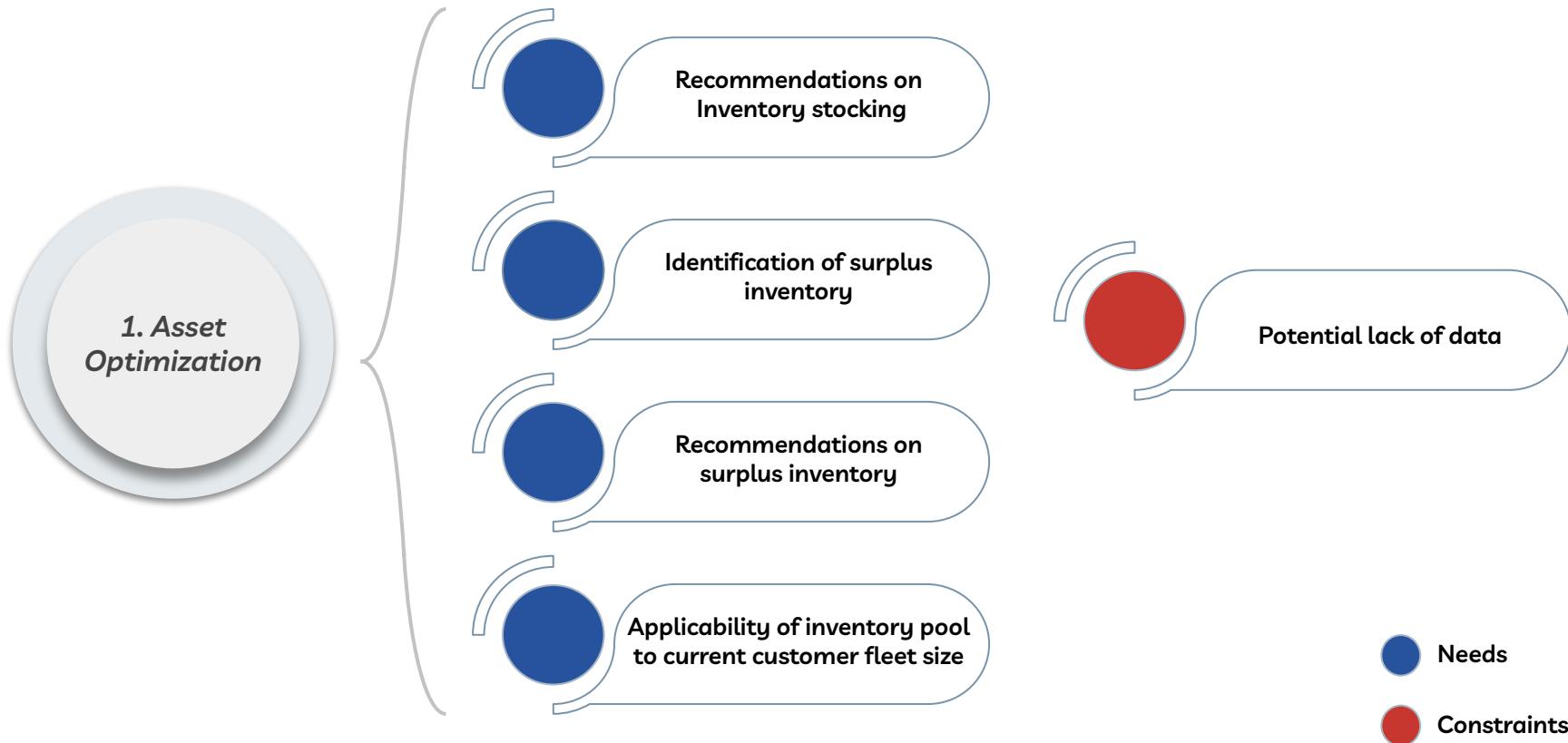


*Rotables are aircraft components which are removed, replace or inspected at intervals and they consist a significant percentage in a typical commercial aircraft.

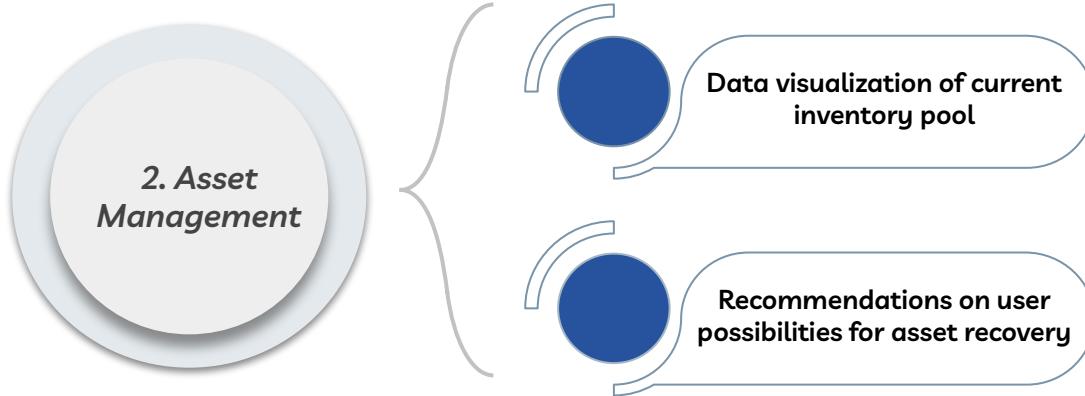
Milestone



User Needs & Constraints

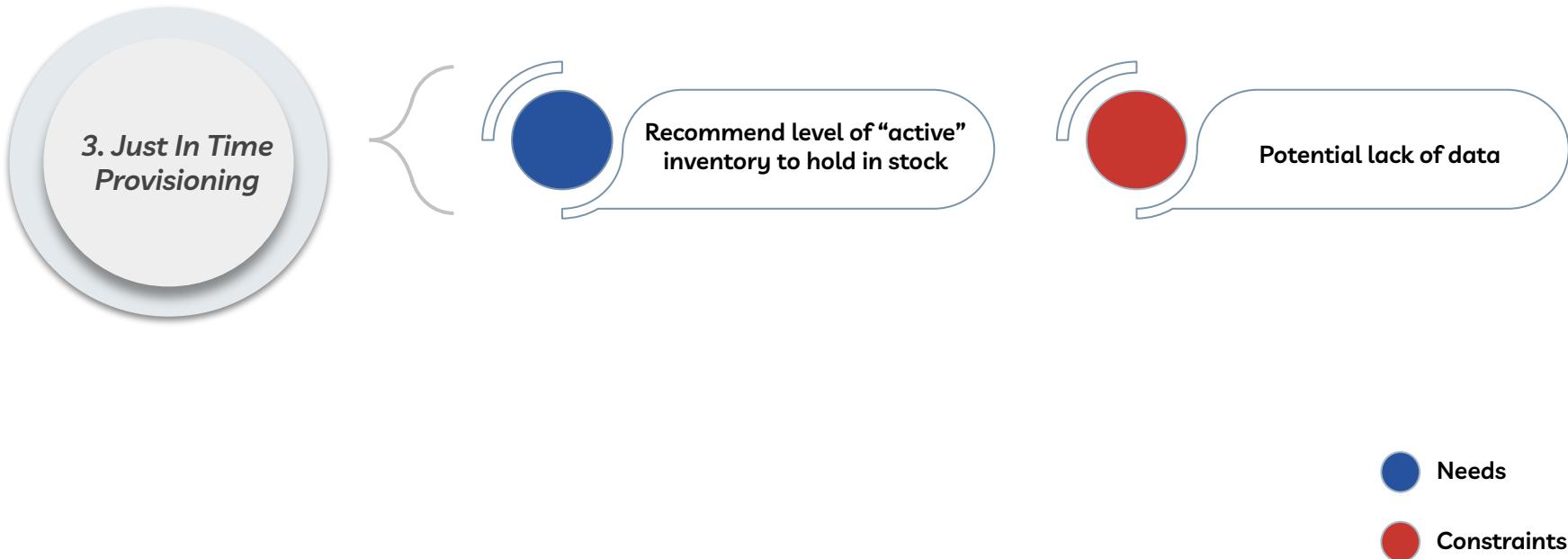


User Needs & Constraints



- **Needs**
- **Constraints**

User Needs & Constraints



User Needs & Constraints



- Needs
- Constraints

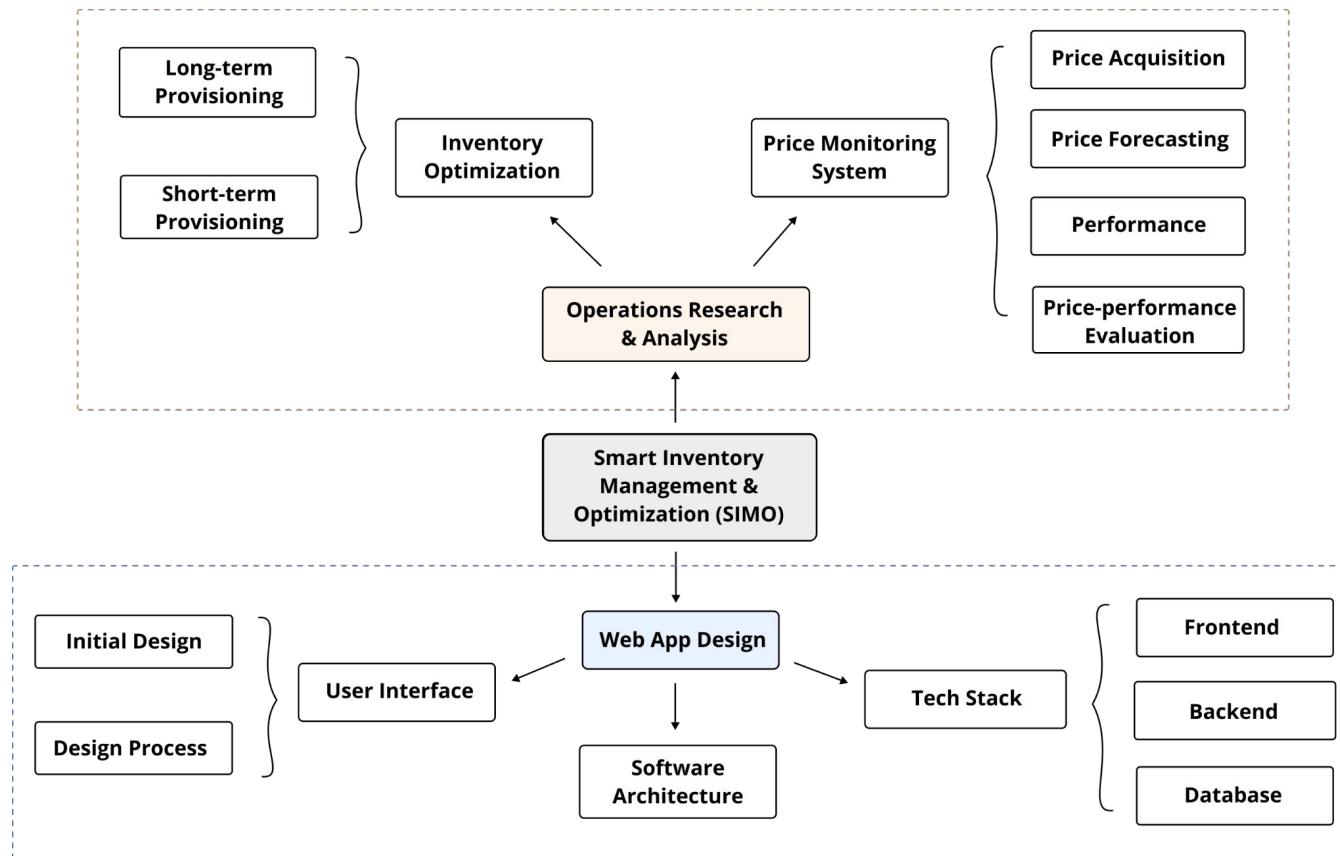
Refined Problem Statement

This project aims to **create a web-based application** with dynamic features such as having **variable inputs, scenario-based optimization** and **data visualization** to help manage and optimize rotables in hopes of a more **user-friendly and cost-saving alternative** for the company to fit into their current workflow.

Design Direction

Concept Generation
Optimizations Research & Analysis
Web Application Design

Concept Generation



Operations Research & Analysis

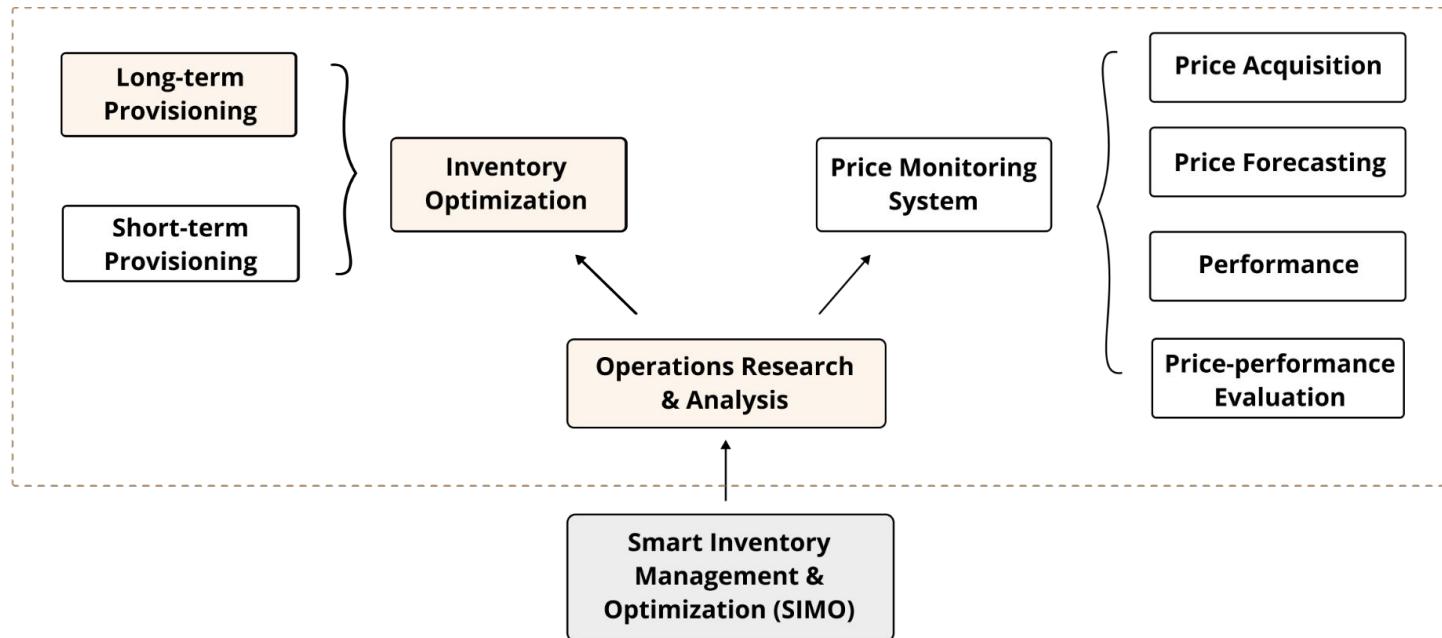
Inventory Optimization

Long-term Provisioning
Short-term Provisioning

Price Monitoring System

Price Acquisition
Performance
Price-performance Evaluation

Long-term Provisioning



Milestone

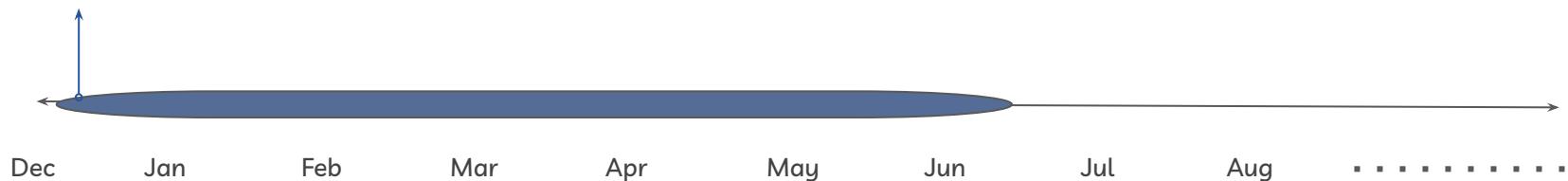


Milestone (Long-term Provisioning)



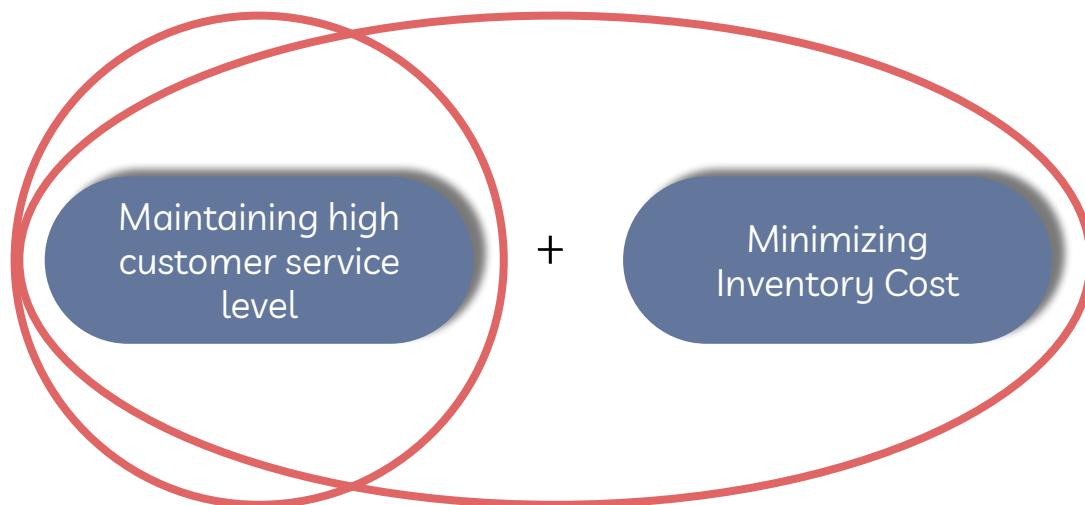
Long-Term Provisioning

Determine Number of Quantity to be topped up 6 months in advance for each type of rotable



Long-Term Provisioning

GOALS:





Approach A

Optimization model

Long-term Provisioning

Approach B

Improving Accuracy & Quality of
Input Data



Approach C

Statistical Distribution of Demand



Approach A

Optimization model

Long-term Provisioning

Approach B

Improving Accuracy & Quality of Input Data

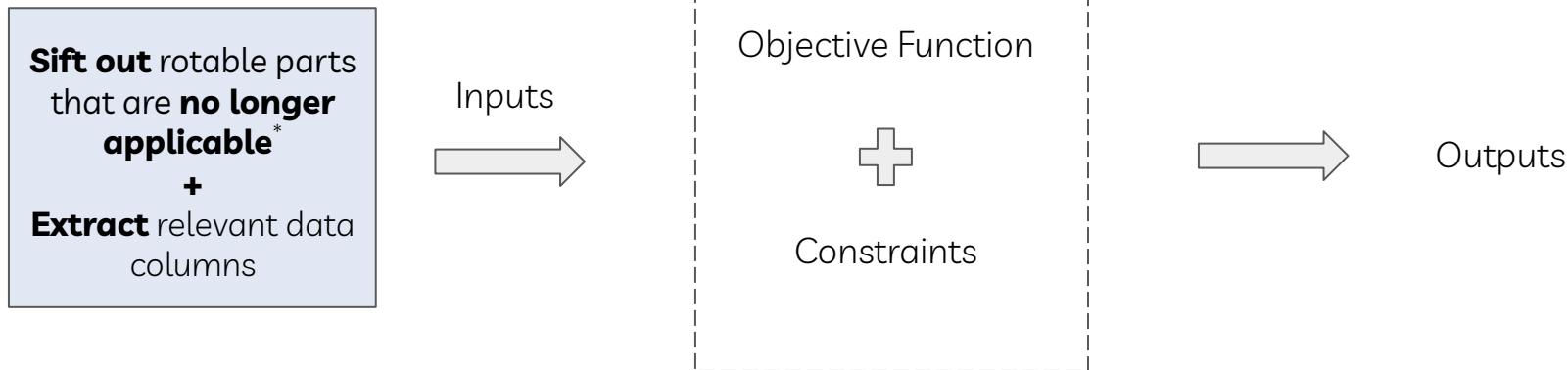


Approach C

Statistical Distribution of Demand

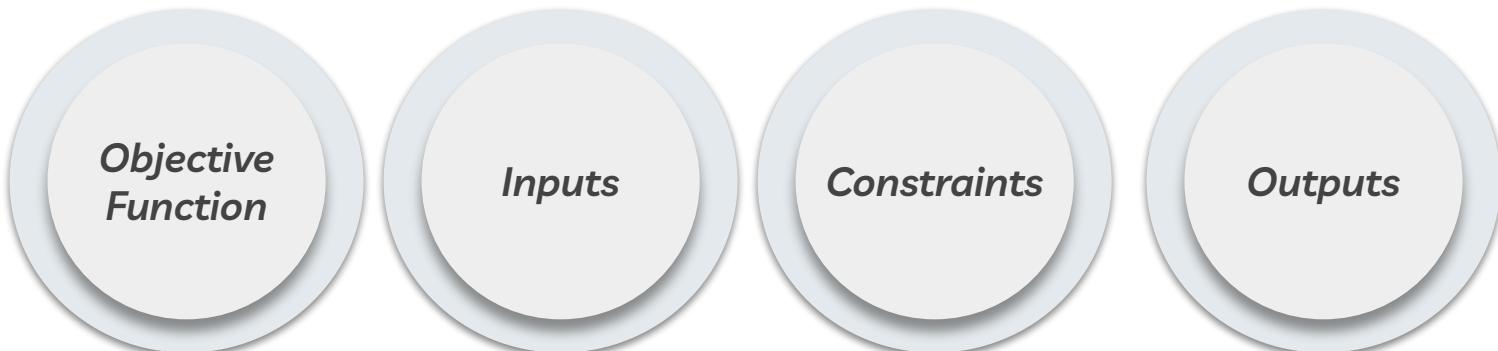
Long-term Provisioning

Optimization Model



* Applicable: Non-applicable rotatable parts are parts that are no longer part of the contractual agreement between SIAEC and their customers hence, SIAEC will not need to provide for these parts to customer

Optimization Model (Formulation)

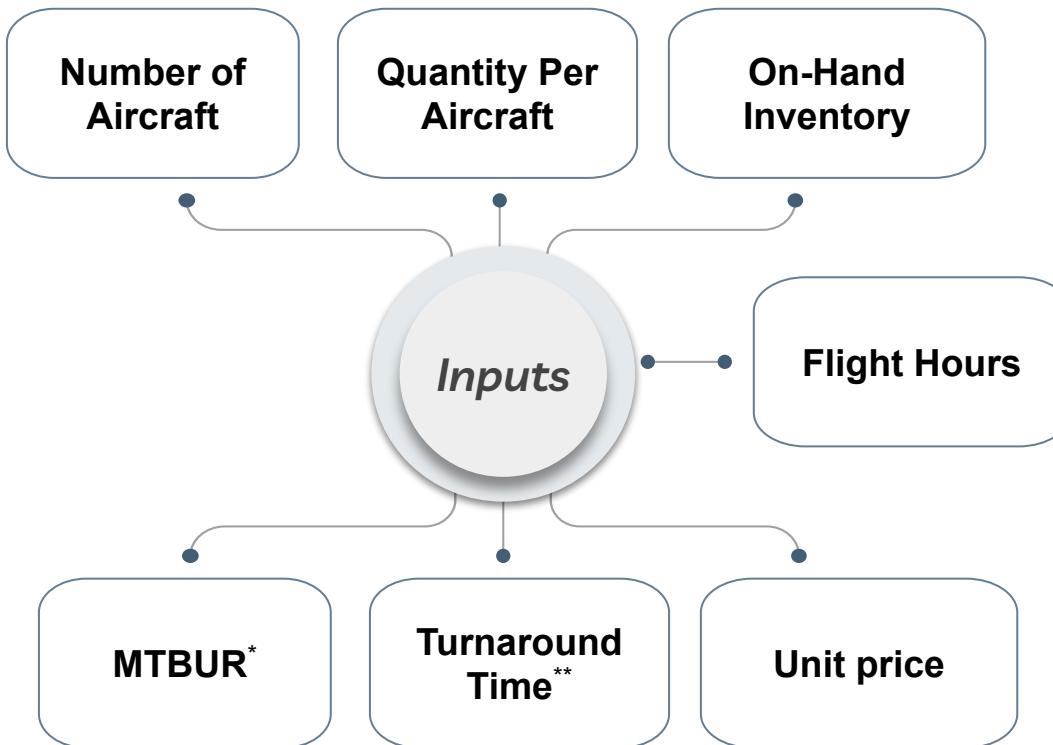


Optimization Model (Objective Function)



- **Minimize** the **total cost** to purchase the required rotables

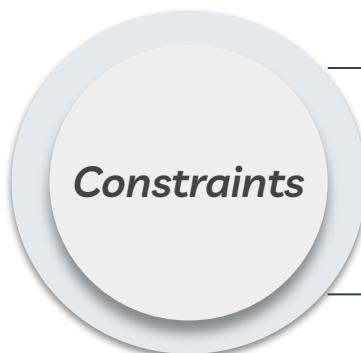
Optimization Model (Inputs)



*MTBUR: Mean Time Before Unscheduled Removal;

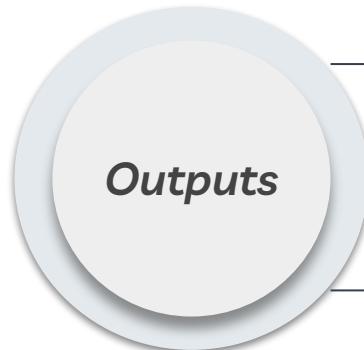
**Turnaround Time includes the repair time and logistics/transportation time

Optimization Model (Constraints)



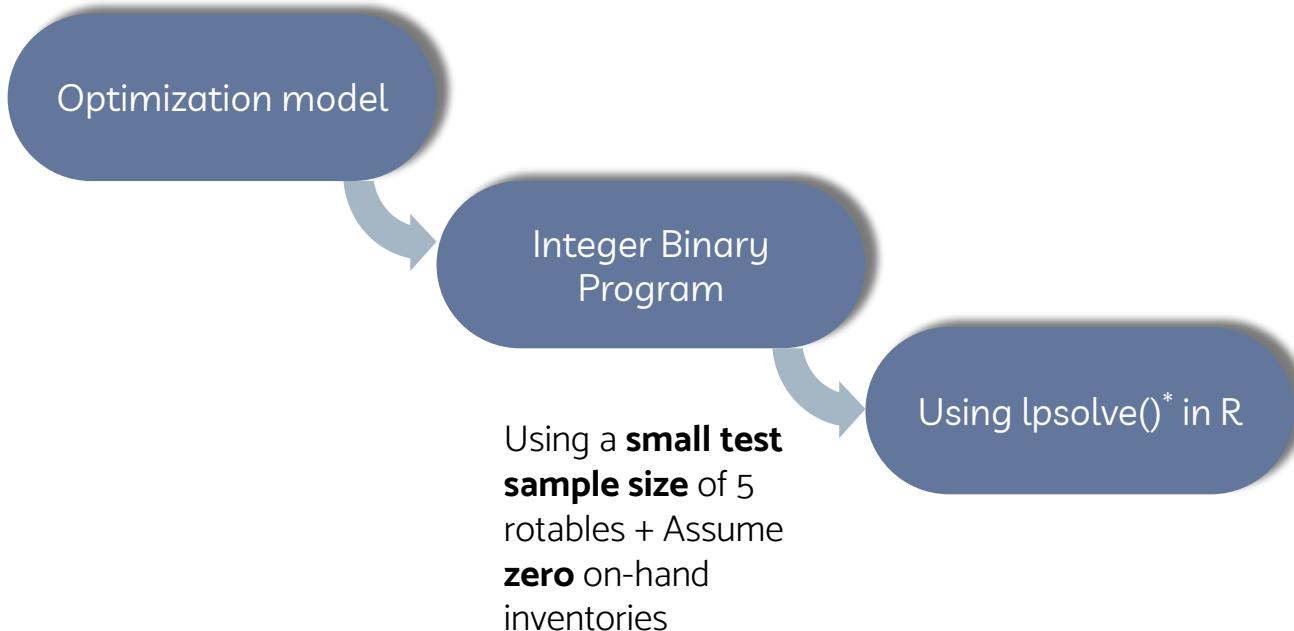
- **Service level** for each class* of rotables are **to be met**
- **Total cost** Incurred should **not exceed** SIAEC budget
- A certain number of specific types of rotables are **always to be available** in the warehouse

Optimization Model (Outputs)



- **Top Up** Quantity for each type of rotable
- Acquired **Service Level** for each class of rotables
- **Total cost** incurred

Optimization Model (Experiment)



Optimization Model (Experiment* result)

Class**	Part Number	Unit Price (\$)	Quantity to hold (SIAEC's method)	Quantity to hold (Our optimization model)
2	12-34	17,712	1	1
2	12-35	17,712	0	0
2	12-36	9501	1	0
2	12-37	4678	1	1
2	12-38	83,971	2	1

SIAEC's current method

Average Service Level: 98.22% (>95%)

Total Investment: USD 199,836

~50%
reduction

Our Optimization model

Average Service Level: 95.82% (>95%)

Total Investment: USD 106,363

*Experiment: To simplify our model, we have assumed on-hand inventory to be zero for this experiment

**Class: Class 2 rotables need to achieve on average a service level of 95%



Approach A

Optimization model

Long-term Provisioning

Approach B

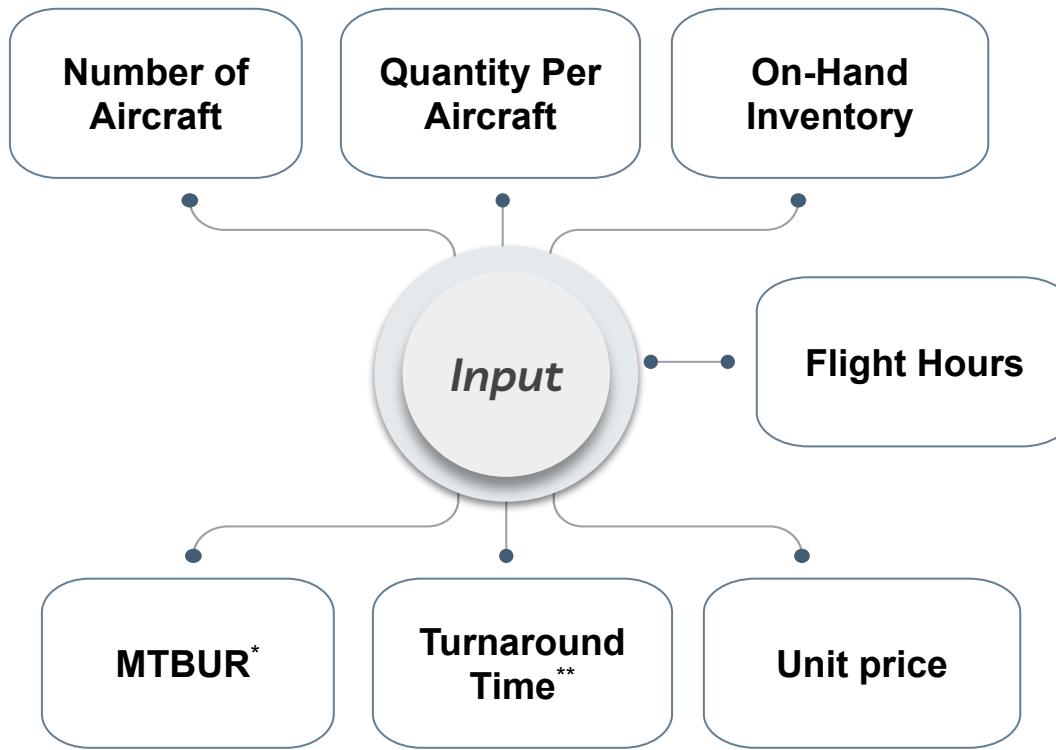
Improving Accuracy & Quality of
Input Data



Approach C

Statistical Distribution of Demand

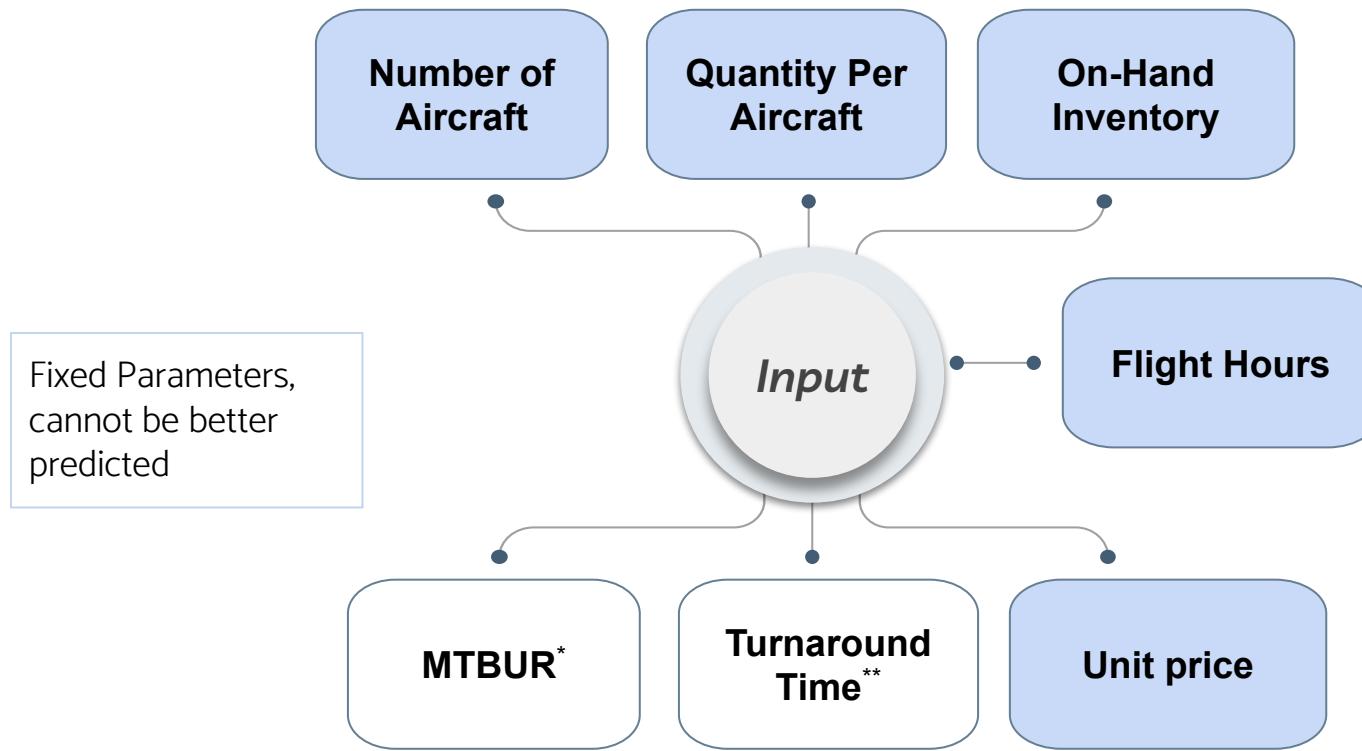
Improving Quality of Input Data



*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

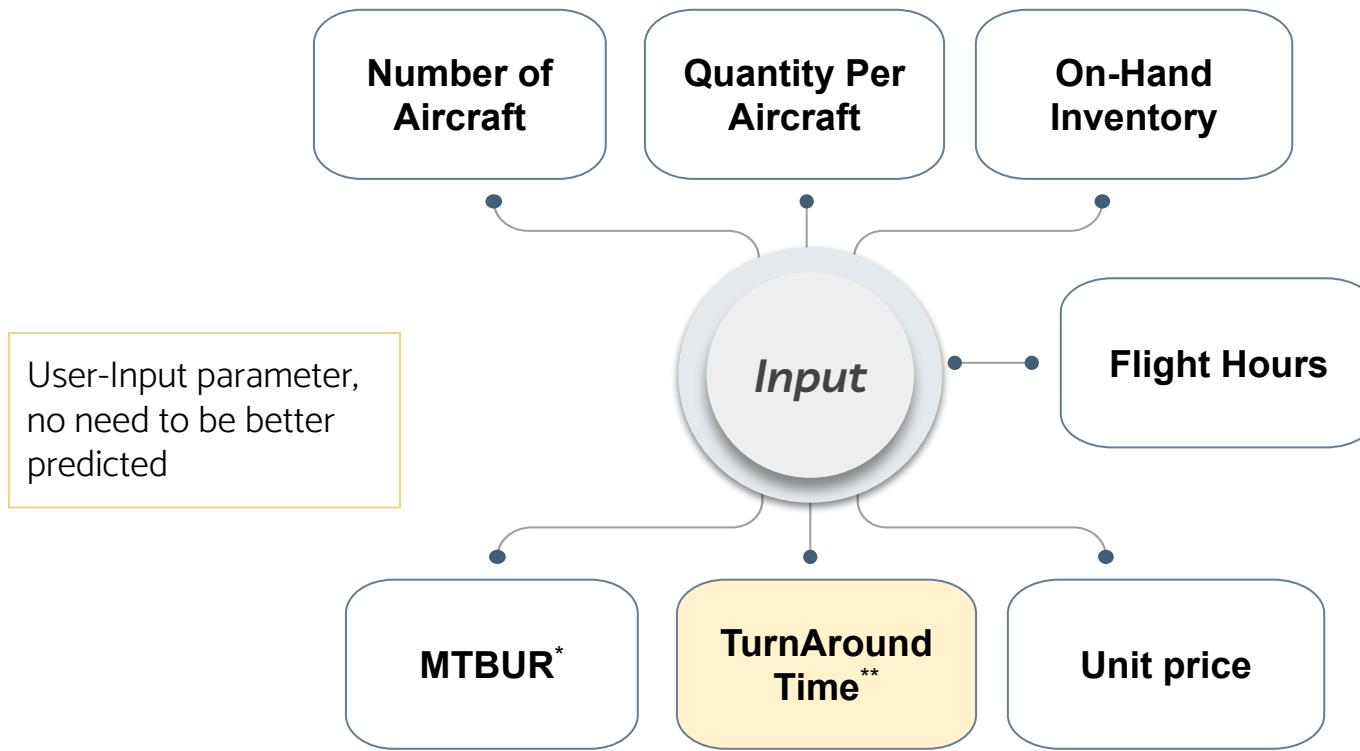
Improving Quality of Input Data



*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

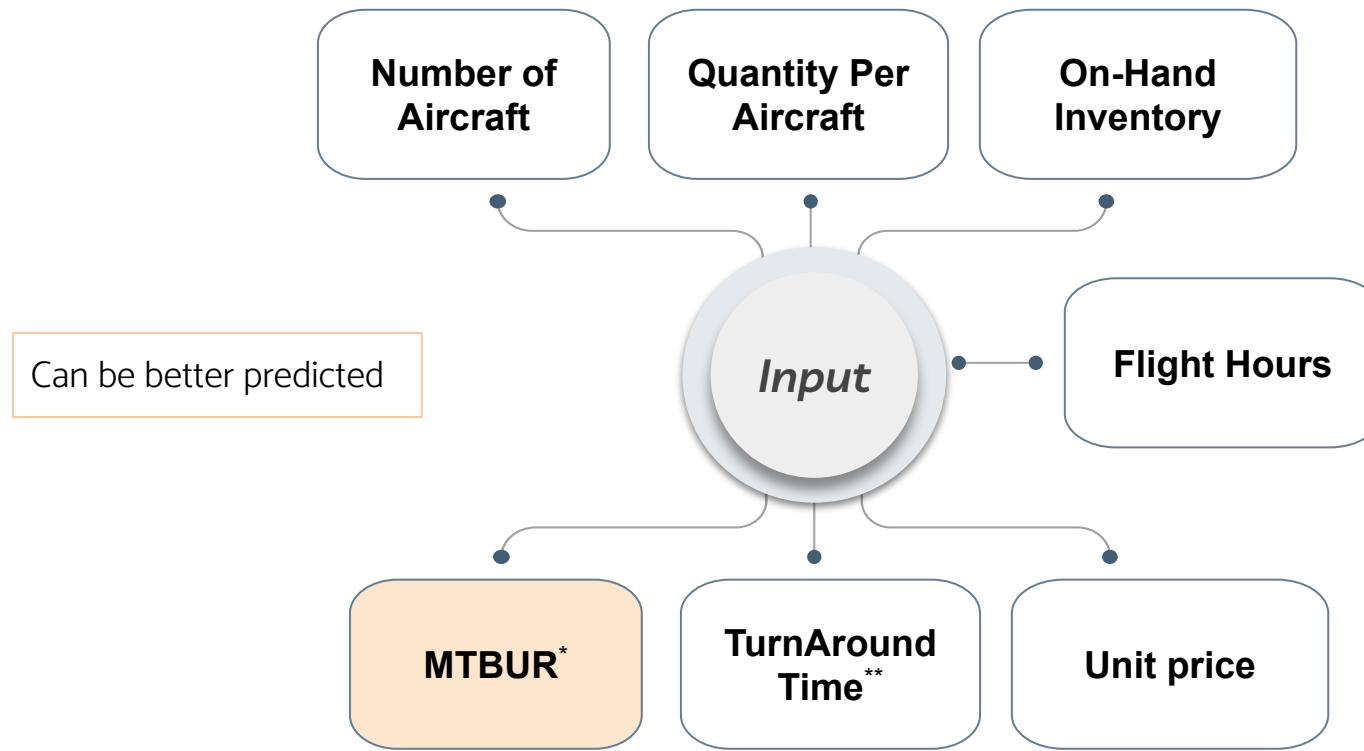
Improving Quality of Input Data



*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

Improving Quality of Input Data

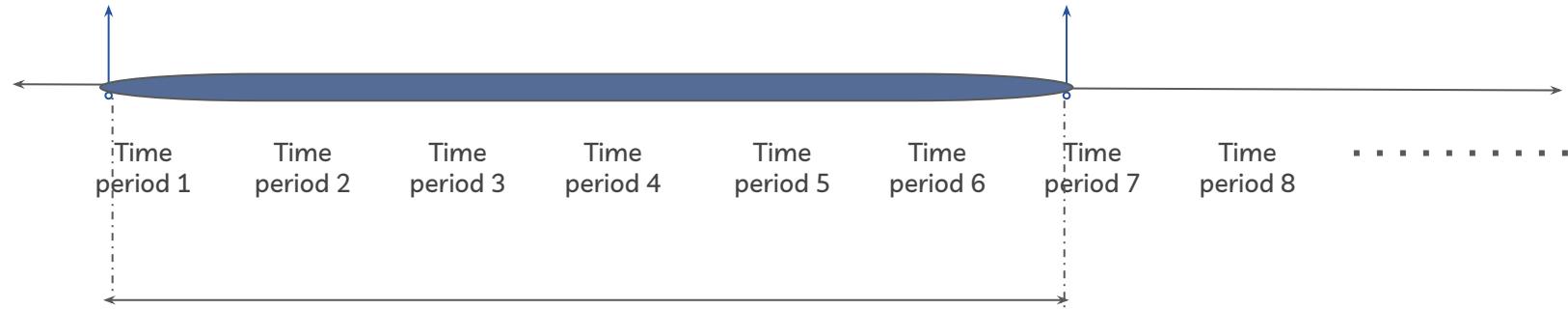


*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

Predicting MTBUR

Rotable i^{th} fitted onto an aircraft



**Mean Time Between
Unscheduled Removal
(MTBUR)**

Each rotatable type will have a MTBUR value of its own

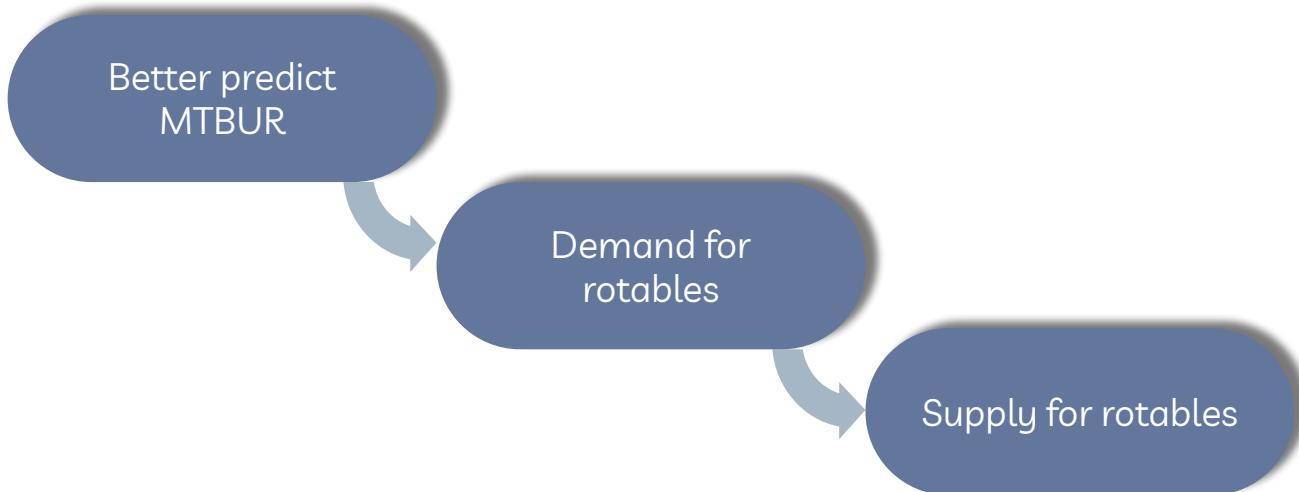
Predicting MTBUR



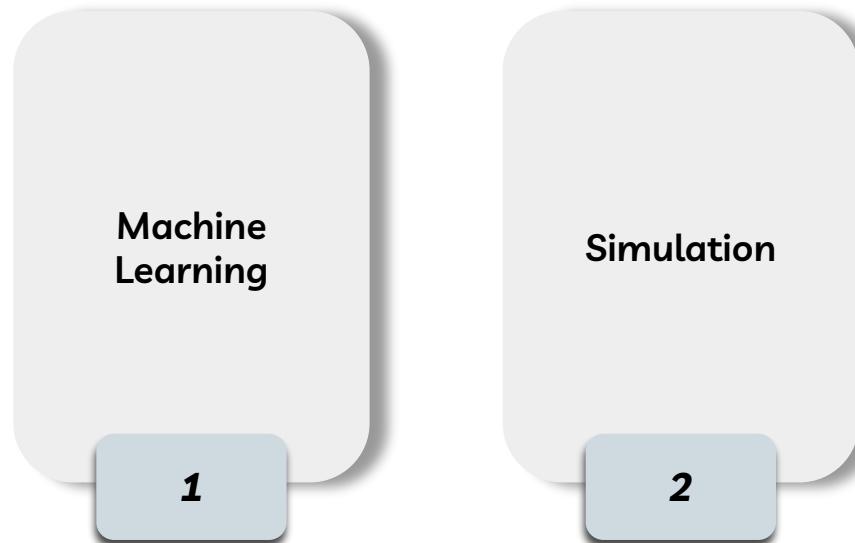
Majority of MTBUR used by SIAEC is provided by the original manufacturer which is **very generalized**¹ and **is skewed**, only a **minority** of rotatable parts have **historical MTBUR** available

¹ Generalized: For example, airplanes that frequently fly to more warm areas might corrode faster (hence lower MTBUR) as compared to cold climates, but these factors affecting the mTBUR are not captured in the aggregated MTBUR that is provided by the original rotatable manufacturer

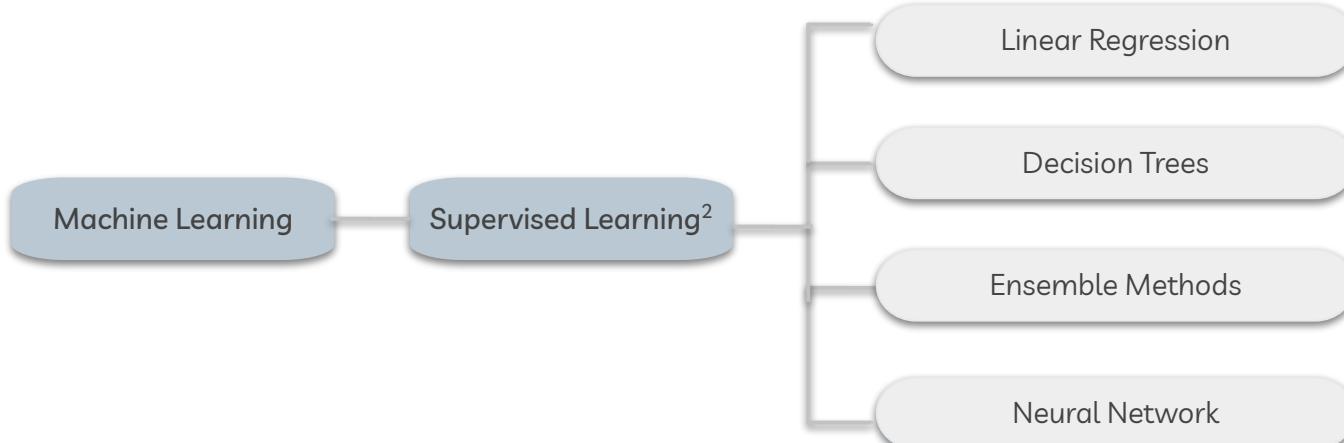
Predicting MTBUR



Predicting MTBUR



Machine Learning



Machine Learning



Factors affecting
MTBUR³

Running Life

Machine usage

History of past failures

Number of past failures

Operating Conditions

Machine Learning



Factors affecting
MTBUR

Running Life

Machine usage

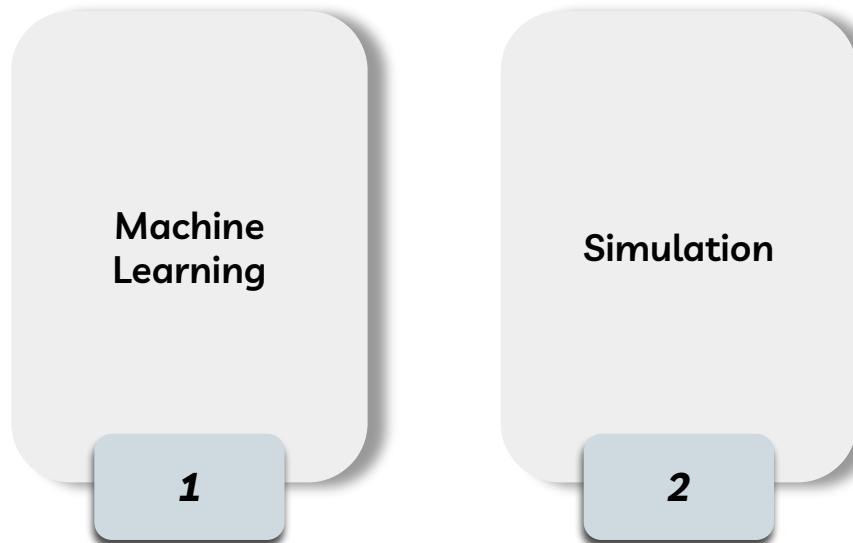
History of past failures

Number of past failures

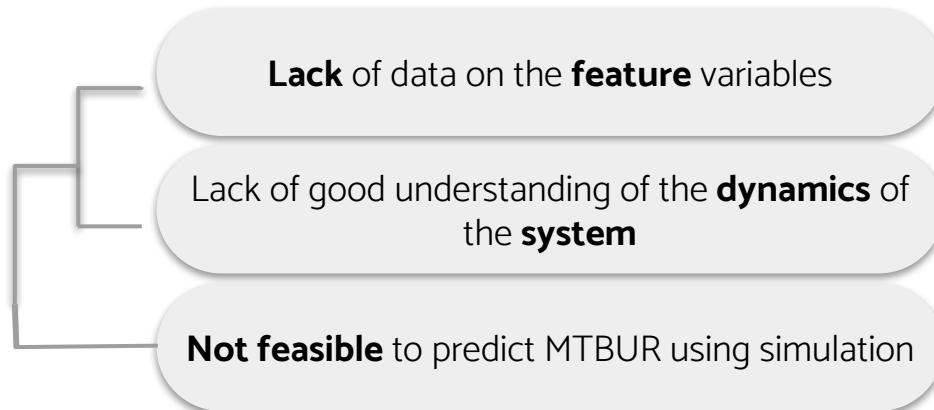
Operating Conditions

Lack of these data

Predicting MTBUR



Simulation⁴





Approach A

Optimization model

Long-term Provisioning

Approach B

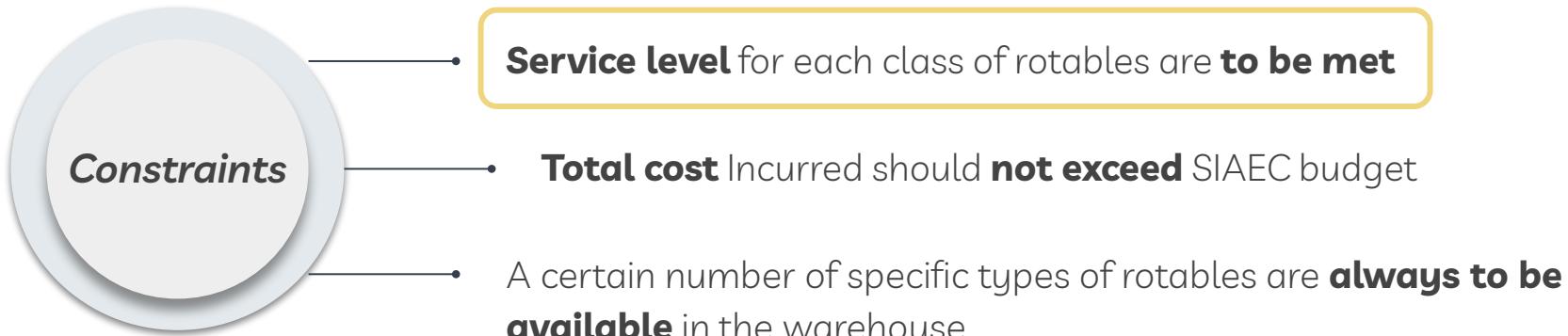
Improving Accuracy & Quality of
Input Data



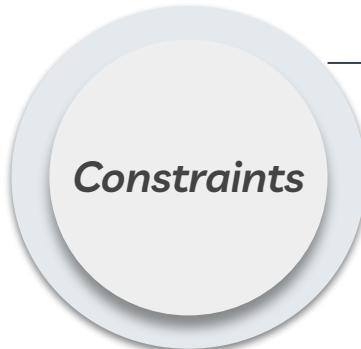
Approach C

Statistical Distribution of Demand

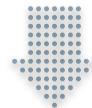
Statistical Distribution of Demand



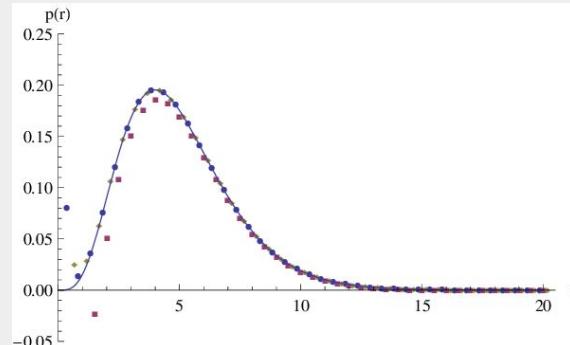
Statistical Distribution of Demand



Service level for each class of rotables are **to be met**



SIAEC assumes Poisson Distribution



Statistical Distribution of Demand

Alternative

Compound

- Stuttering Poisson
- Negative Binomial

Non-Compound

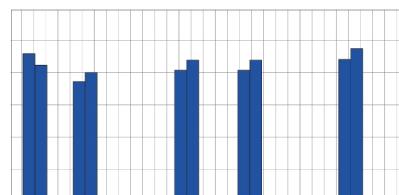
- Normal
- Gamma

Statistical Distribution of Demand

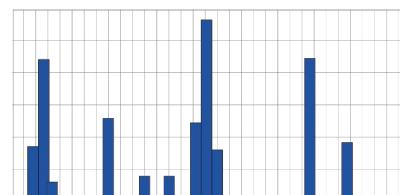
Average Demand Interval (ADI)

$CV^2 = 0.49$

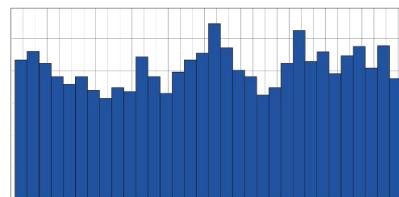
Intermittent



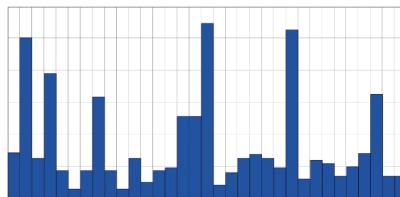
Lumpy



Smooth



Erratic

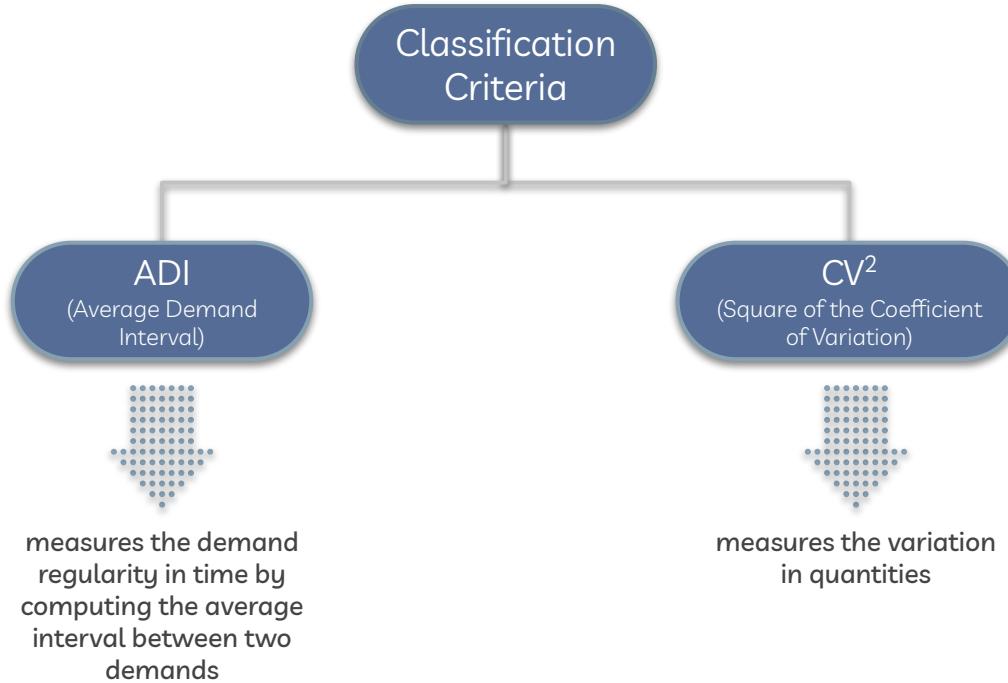


ADI = 1.32

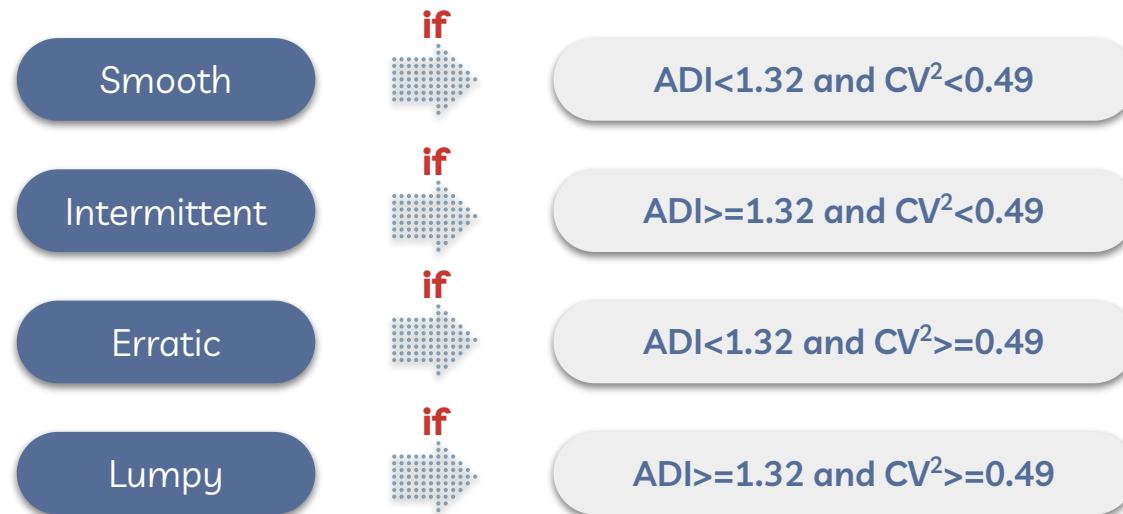
Figure 1: Demand Classification⁵.

Squared of Coefficient of Variation (CV^2)

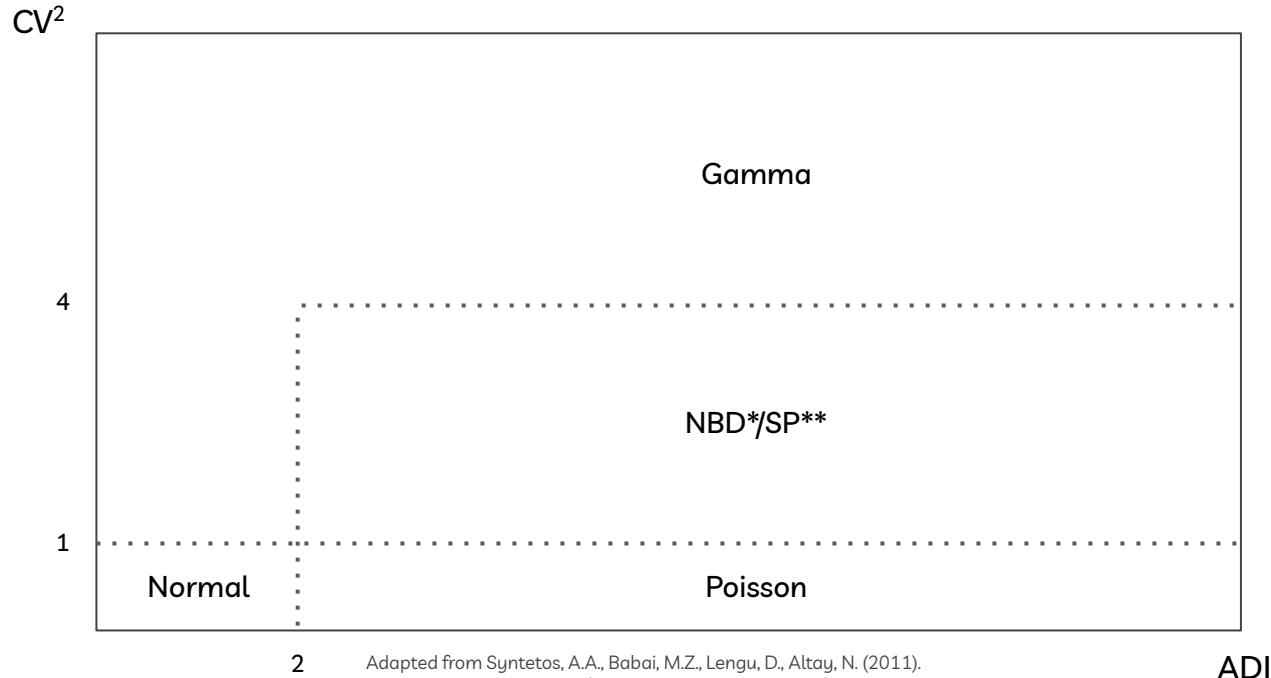
Statistical Distribution of Demand⁶



Statistical Distribution of Demand⁶



Statistical Distribution of Demand



* Negative Binomial Distribution

** Stuttering Poisson

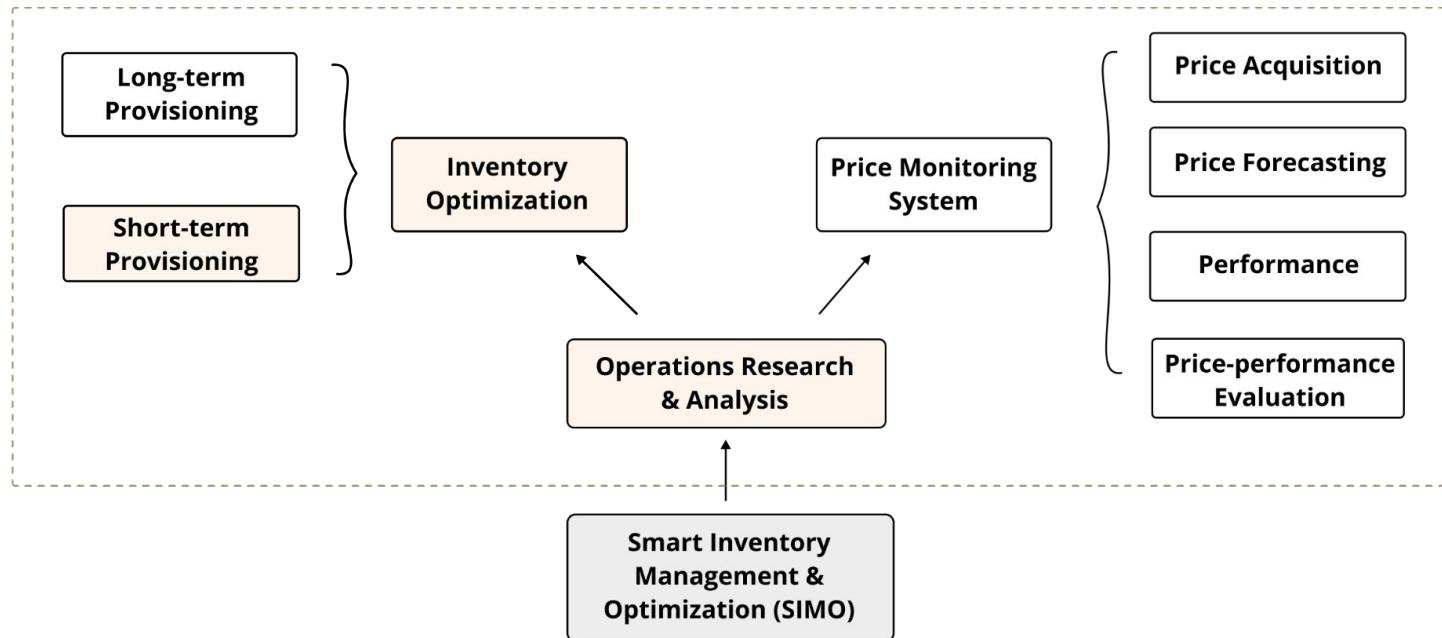
2 Adapted from Syntetos, A.A., Babai, M.Z., Lengu, D., Altay, N. (2011).
Distributional Assumptions for Parametric Forecasting of Intermittent
Demand. In: Altay, N., Litteral, L. (eds) Service Parts Management.
Springer, London. Distributional Assumptions for Parametric
Forecasting of Intermittent Demand | SpringerLink



Milestone



Short-term Provisioning

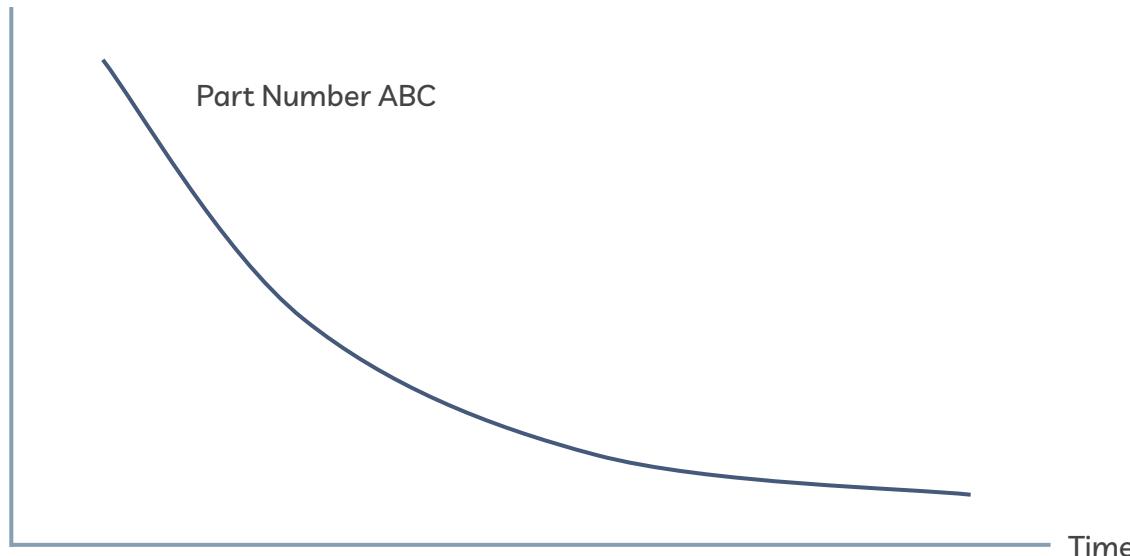


Milestone

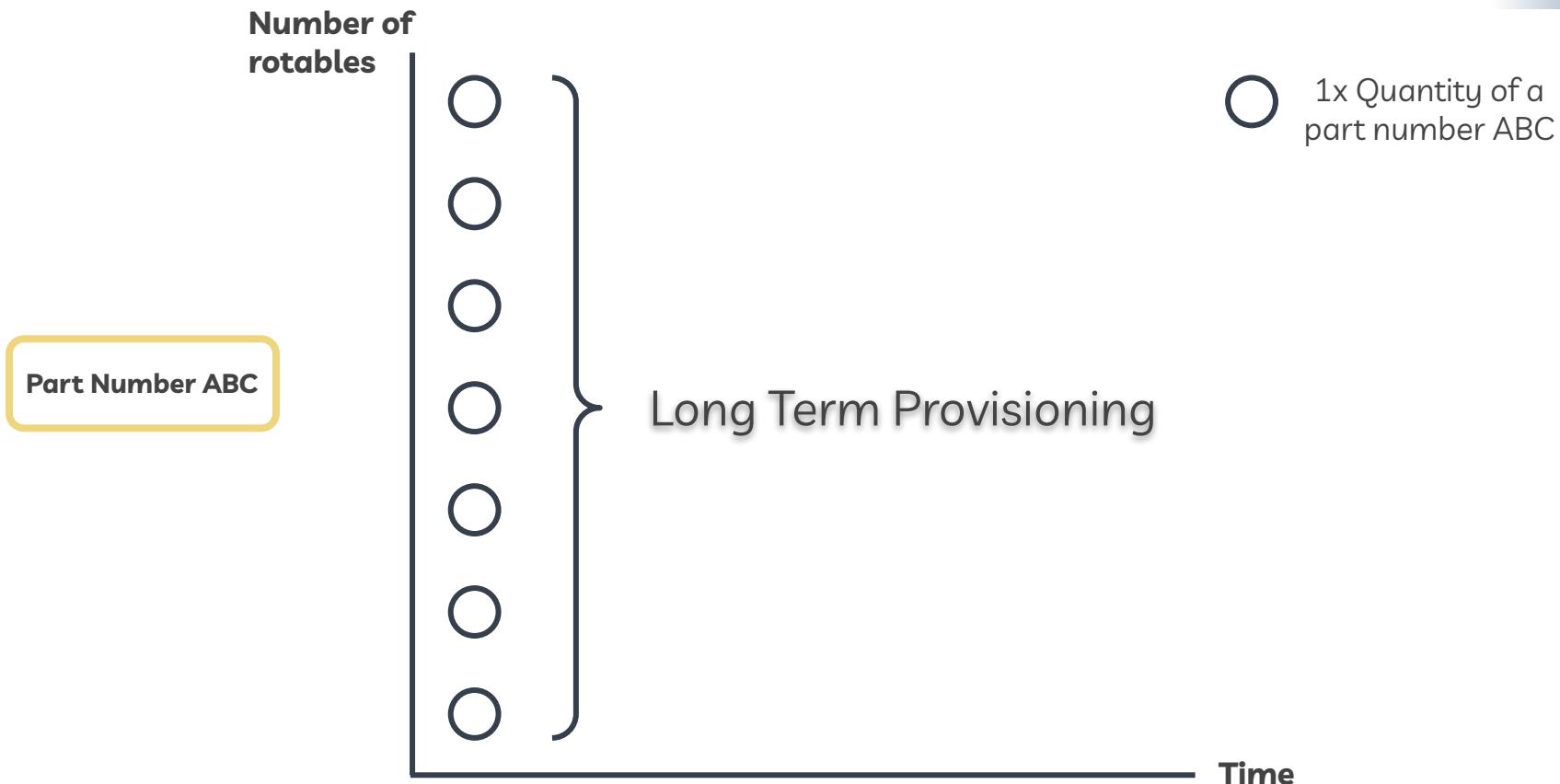


Short-term Provisioning

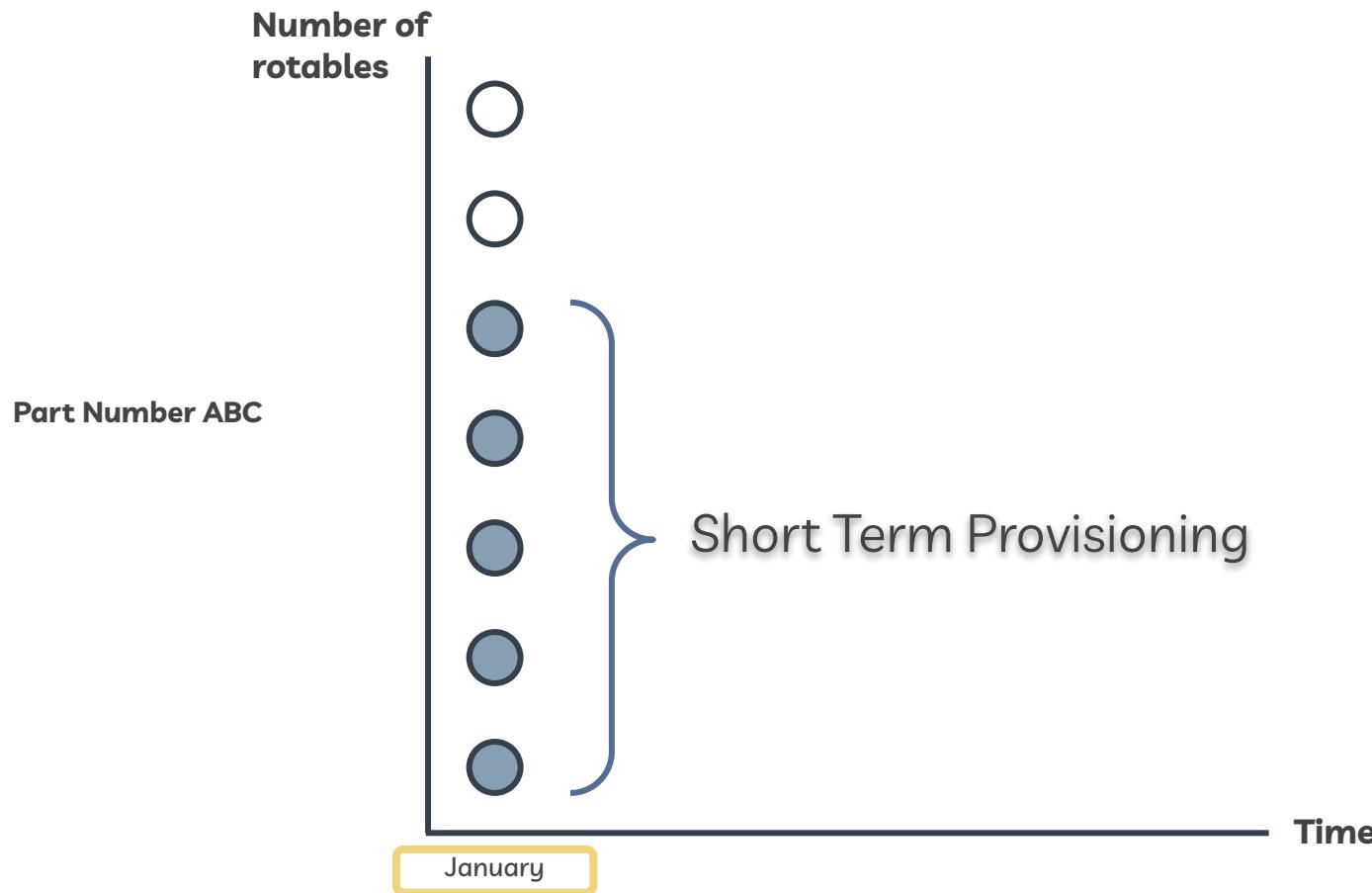
Rotables Condition



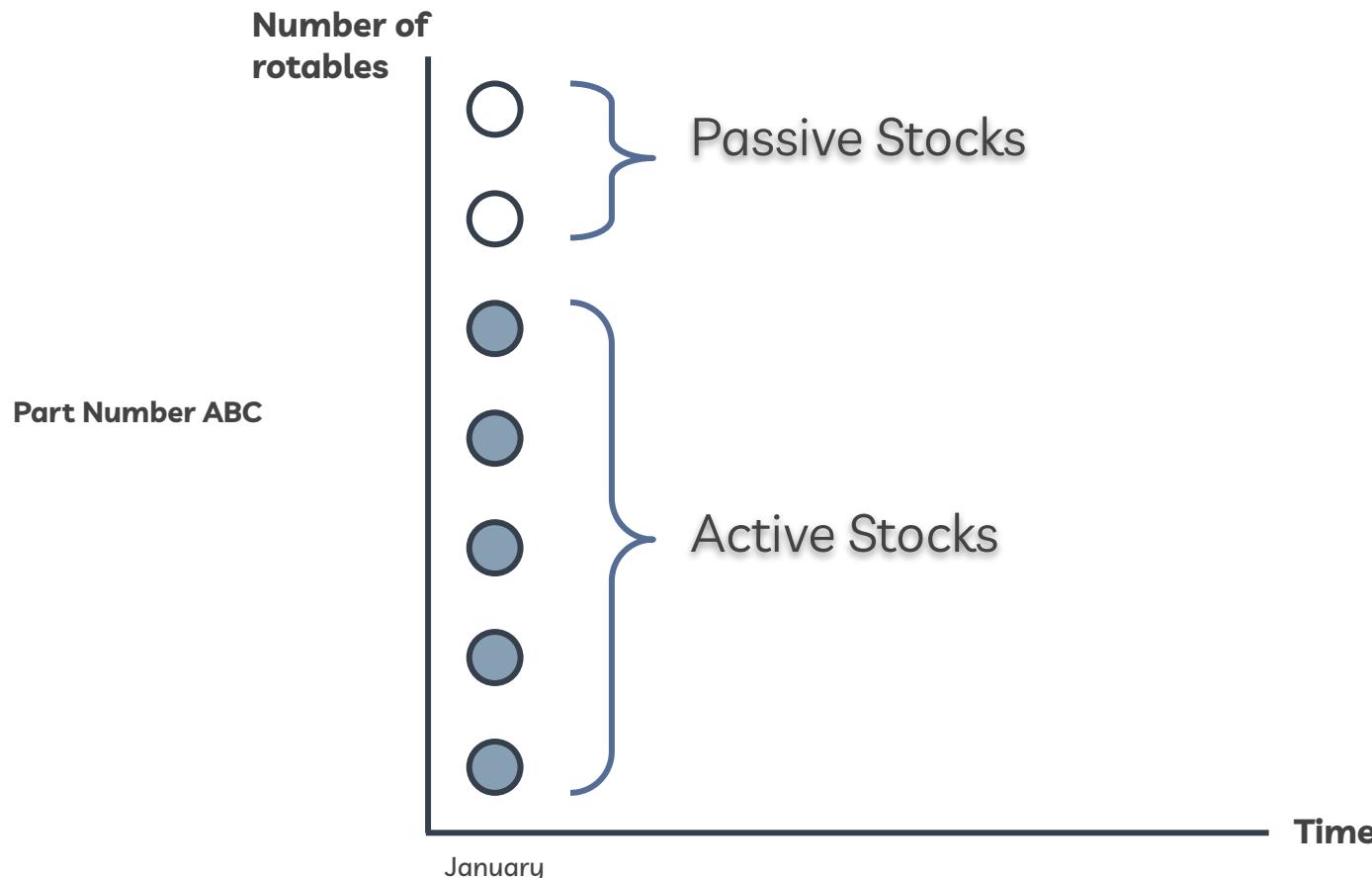
Short-term Provisioning



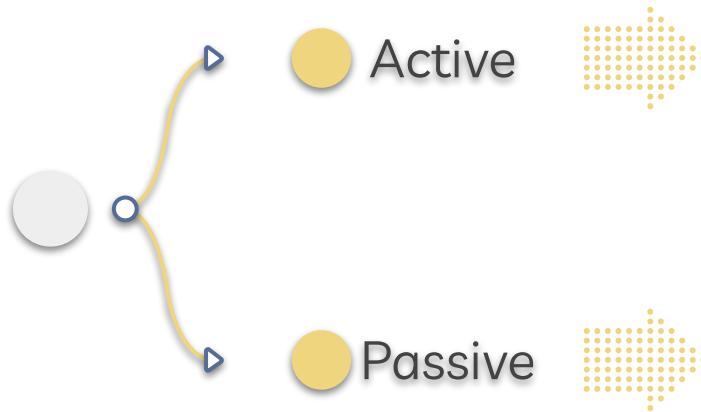
Short-term Provisioning



Short-term Provisioning



Short-term Provisioning



- Freshly Repaired
 - Optimal Condition (Tip-top condition)
-
- To preserve rotables
 - Slow down the deterioration rate

Short-term Provisioning

So what are our approaches to forecasting for active rotables?

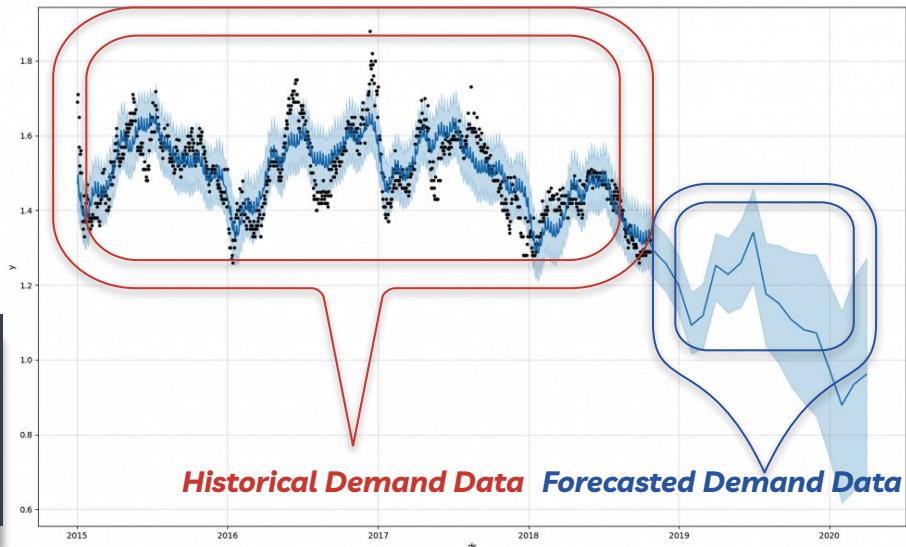
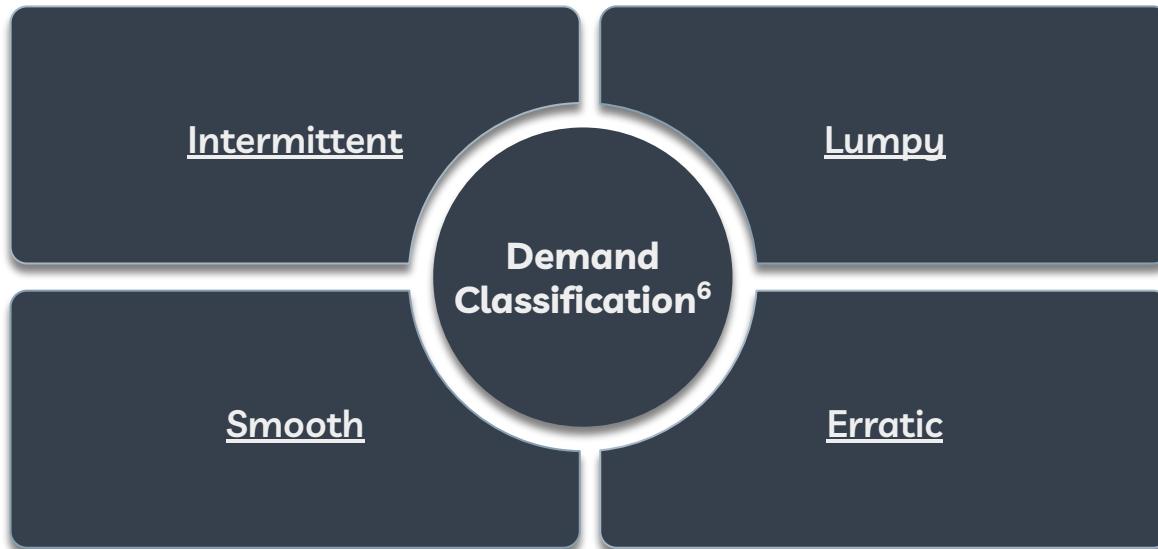
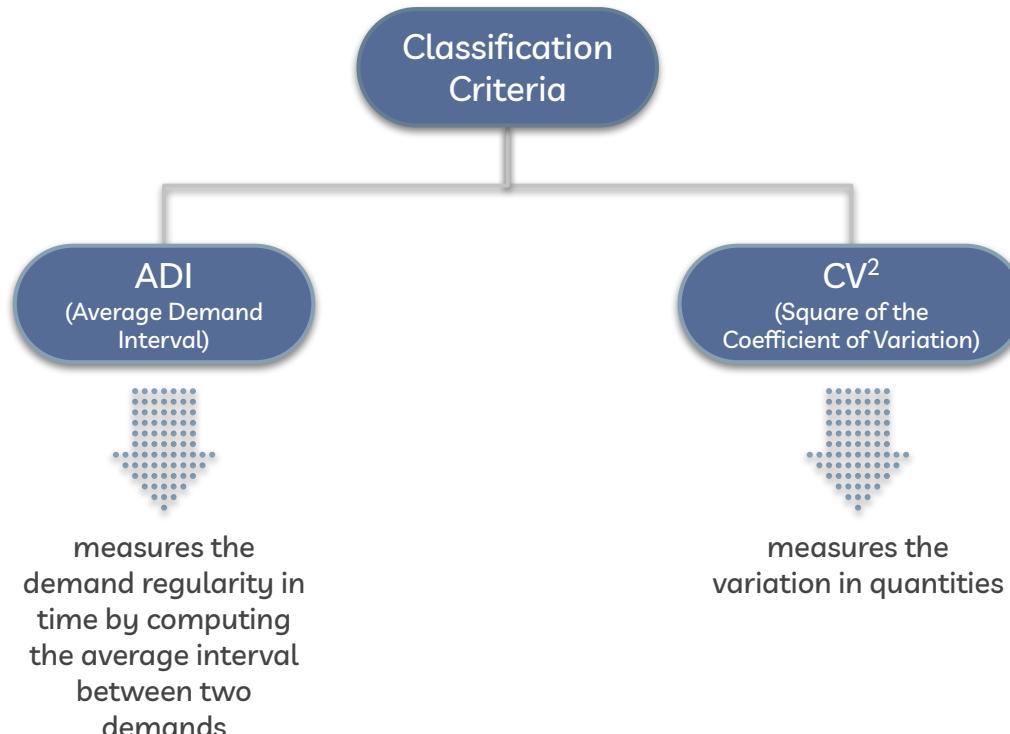


Figure 2: Time Series Forecasting. Adapted from Colin.catlin. (n.d.). Syllepsis. Retrieved April 12, 2022, from <https://syllepsis.live/2019/10/08/time-series-forecasting-a-quick-reference/>

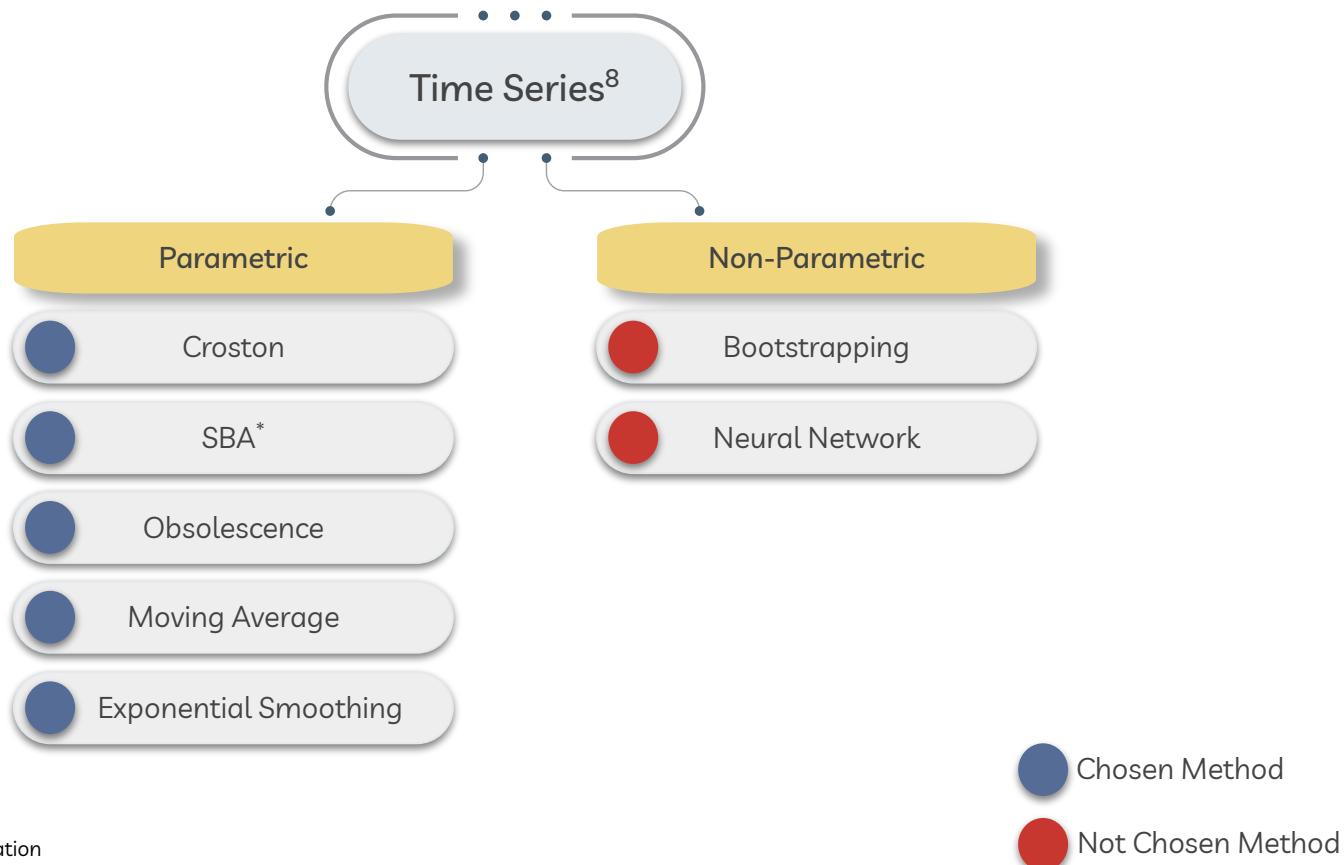
Short-term Provisioning



Short-term Provisioning



Short-term Provisioning



Short-term Provisioning

Design Criteria	Alternative Design Concepts	
	Parametric	Non-Parametric
Data Requirement	+	0
Accuracy	0	+
Time	+	0
Ease of implementation	+	0
Totals	+++	+

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.1

Price Monitoring System



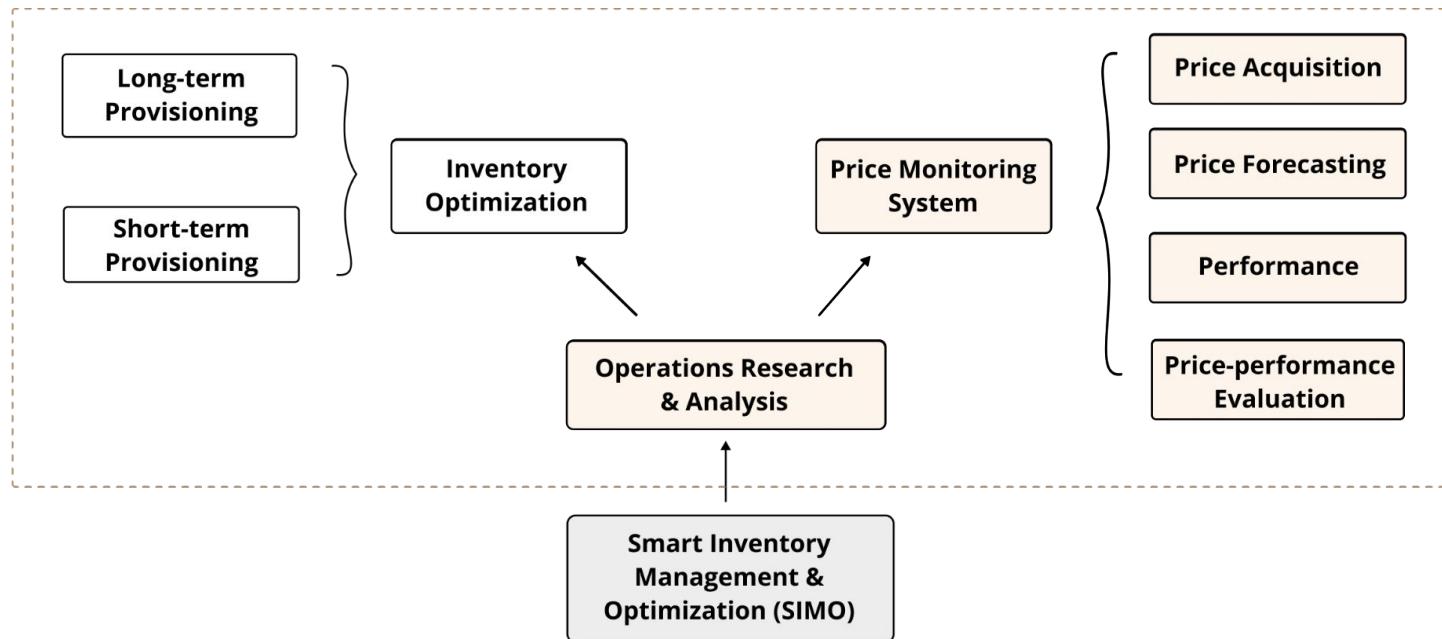
Short-term Provisioning



Short-term Provisioning



Price Monitoring System (Price Acquisition)



Price Monitoring System

- Price monitoring or price intelligence refers to the awareness of pricing in the market and the response to these changes in pricing.
- Metasearch engine⁹ → Sending queries through multiple search engines and aggregating these results
 - Ranking
 - Combine



Price Monitoring System



Price
Acquisition



Price
Forecasting



Performance
Evaluation

Price Monitoring System



Price
Acquisition

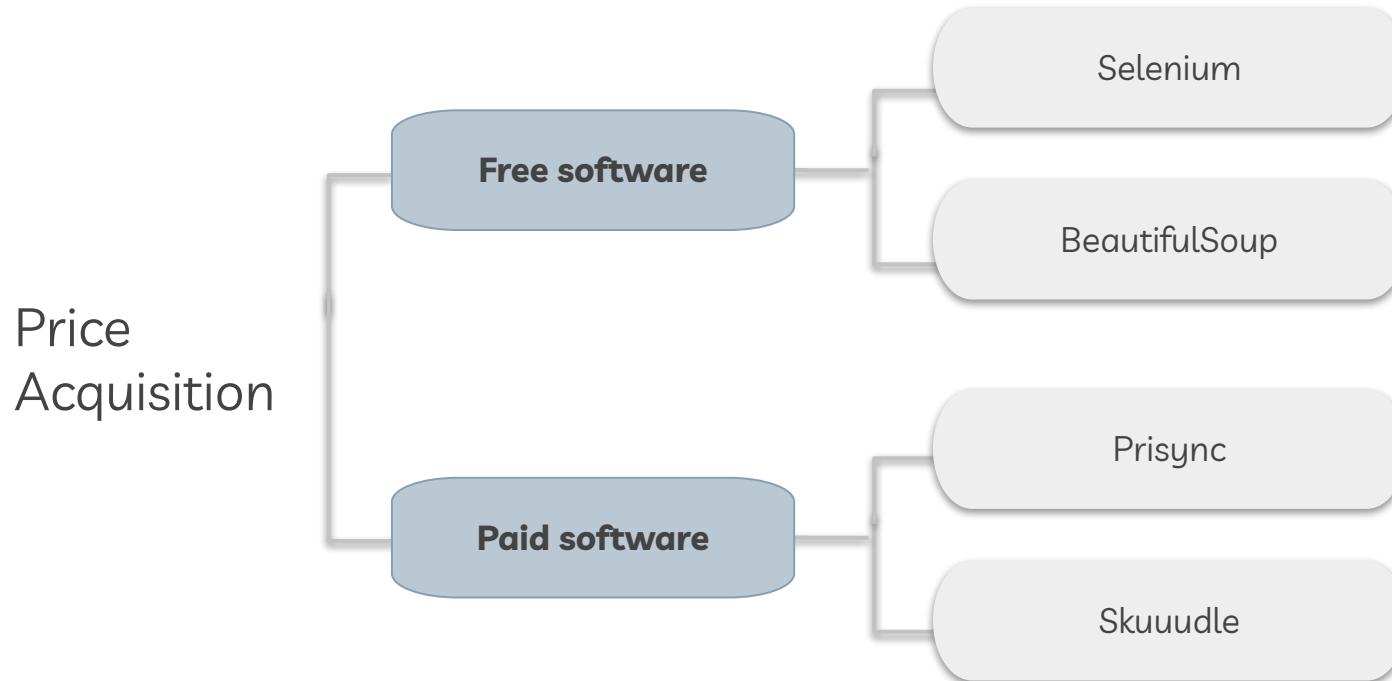


Price
Forecasting



Performance
Evaluation

Price Monitoring System



Price Monitoring System (Price Acquisition)

Design Criteria	Alternative Design Concepts			
	Skuuuudle	Prisync	BeautifulSoup	Selenium
Affordability	-	0	+	+
Ease of Integration	-	0	+	+
Adaptability	+	+	-	0
Total	-	+	+	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.2



Selenium

Price Monitoring System



Price
Acquisition

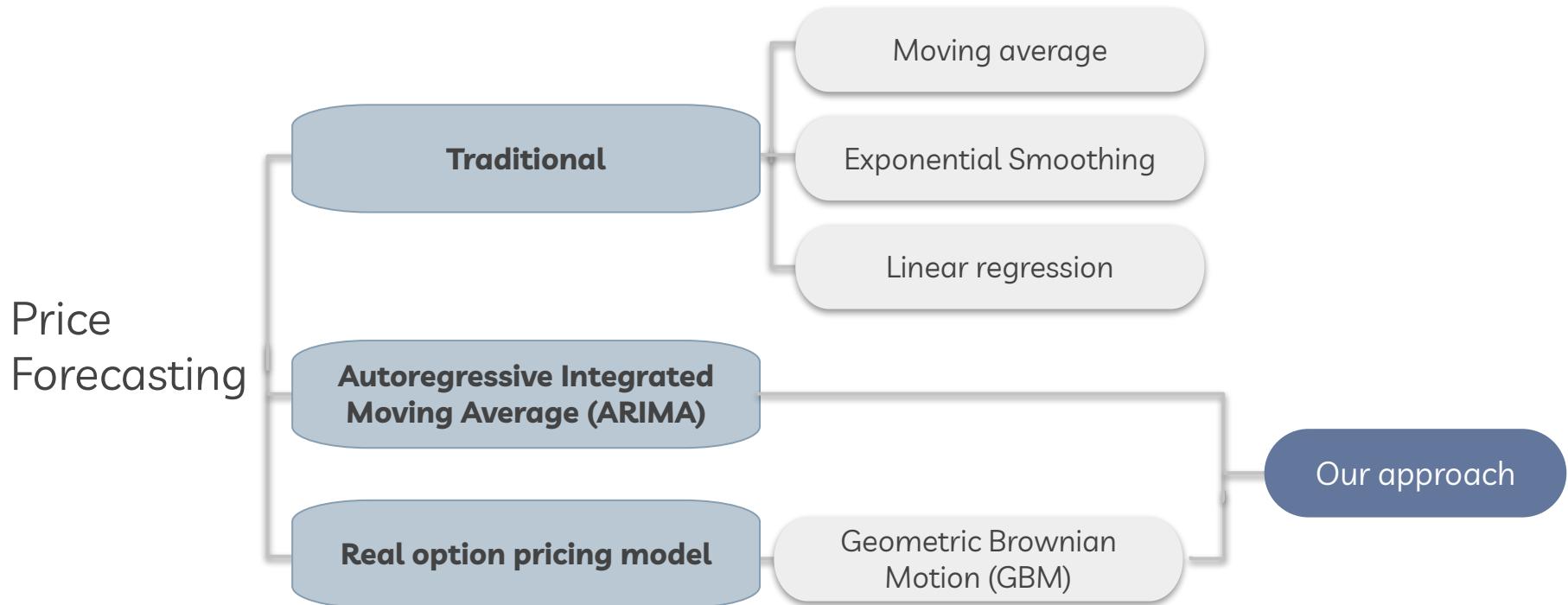


Price
Forecasting



Performance
Evaluation

Price Monitoring System (Price Forecasting)¹⁰



Price Monitoring System (Price Forecasting)¹⁰

Design Criteria	Alternative Design Concepts	
	Geometric Brownian Motion	ARIMA
Data Requirement	+	+
Accuracy	+	0
Time	0	0
Ease of implementation	+	+
Totals	+++	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.3

Price Monitoring System



Price
Acquisition

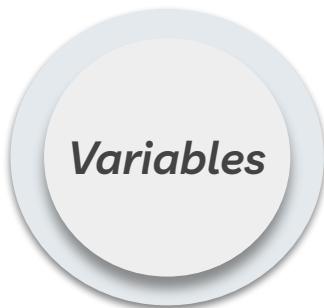


Price
Forecasting



Performance
Evaluation

Price Monitoring System (Performance)

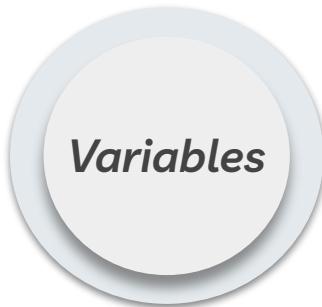


Time Lasted on Wing (TSI) across each vendor

Original Equipment Manufacturer (OEM) MTBUR

$$\frac{TSI}{MTBUR}$$

Price Monitoring System (Price to Performance)



MTBUR v.s Time Lasted on Wing (TSI)

Price given by OEMs v.s. Prices quoted

$$\frac{MTBUR}{\text{Price given by OEMs}}$$

Benchmark

$$\frac{TSI}{\text{Price quoted}}$$

Price Monitoring System



Price
Acquisition

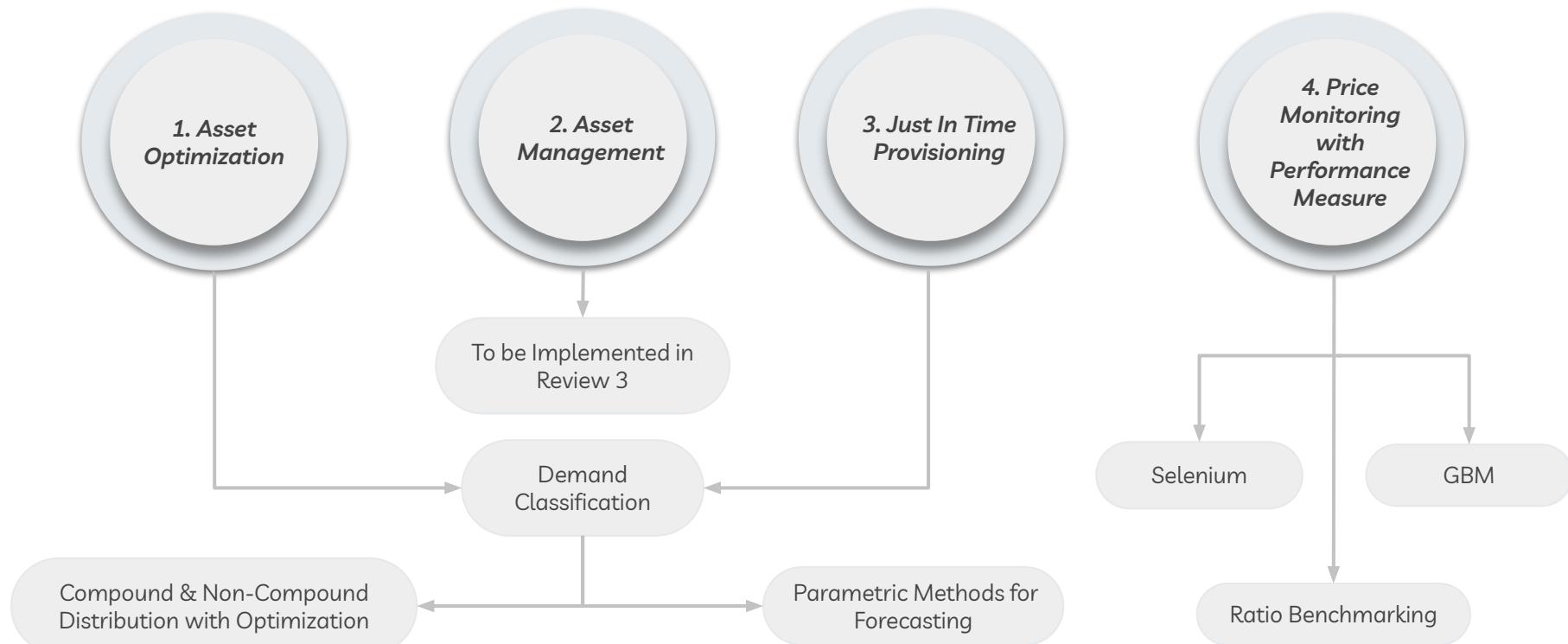


Price
Forecasting

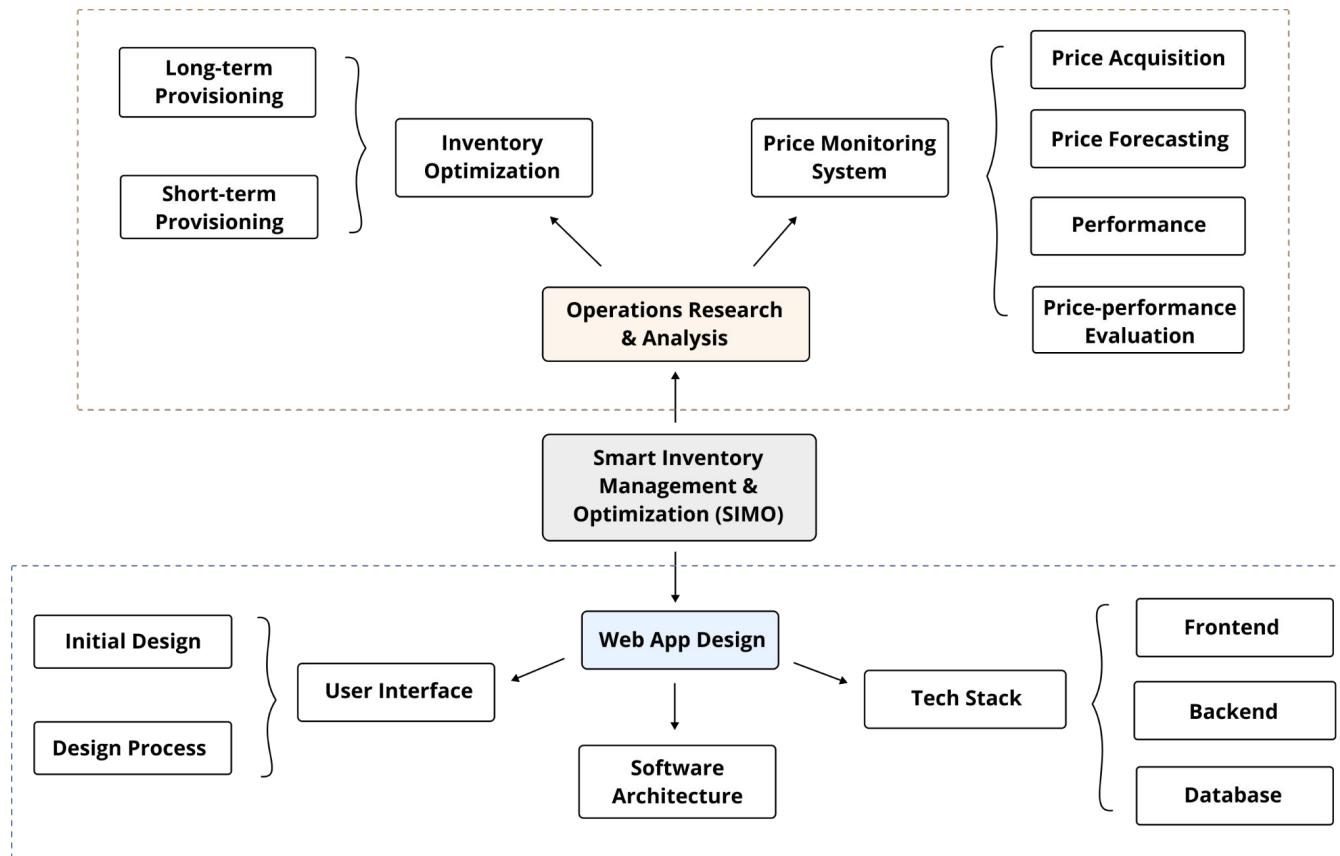


Performance
Evaluation

Summary



Concept Generation



Web Application Design

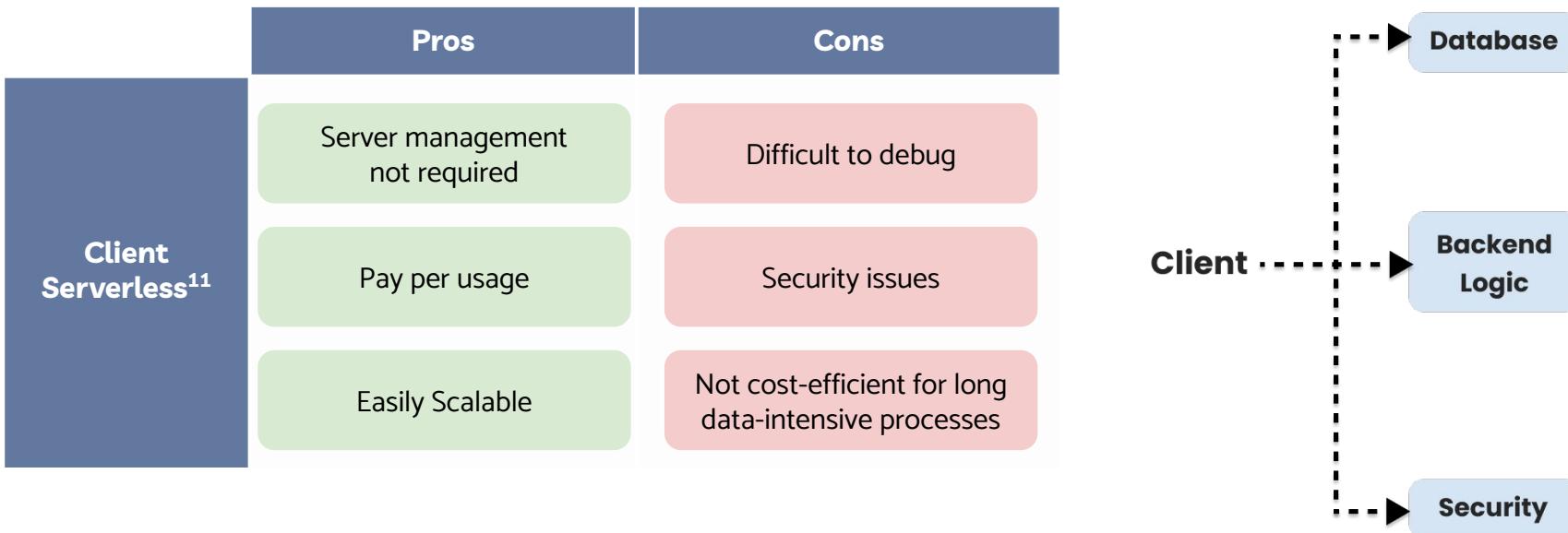
Software Architecture
User Interface
Tech stack

Software Architecture

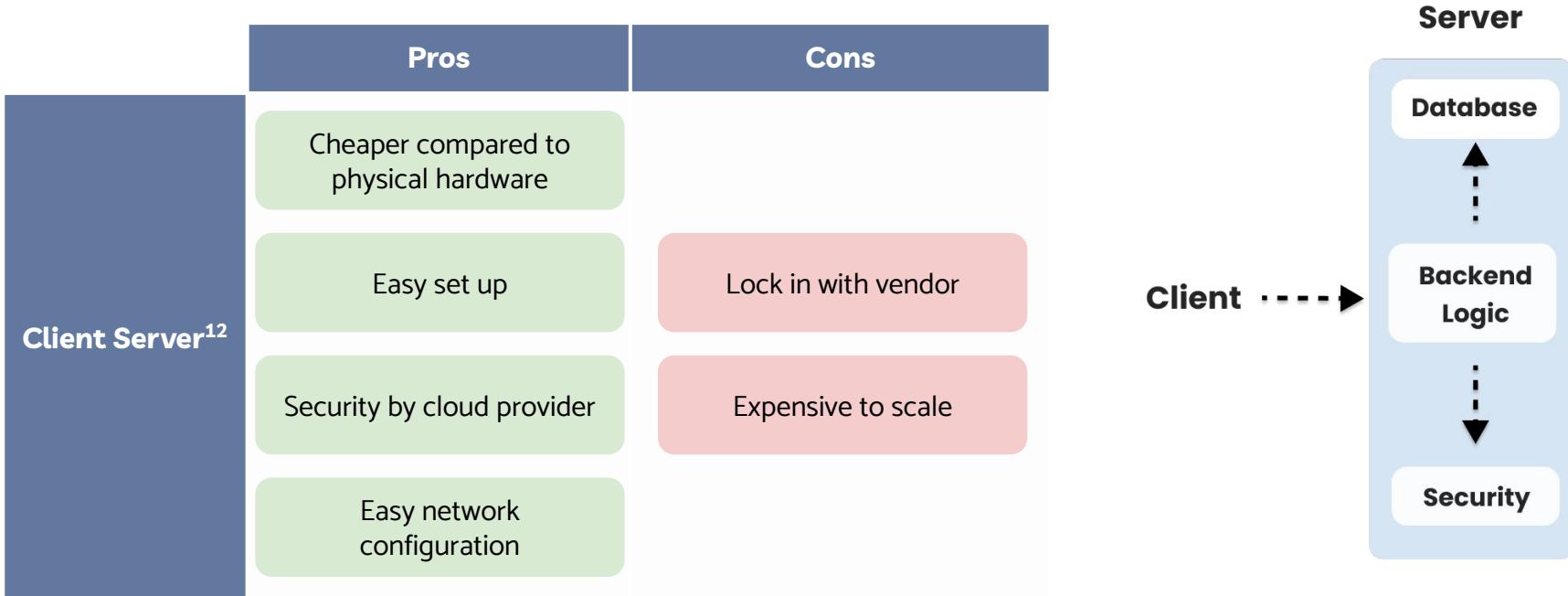
Client Serverless

Client Server

Software Architecture



Software Architecture

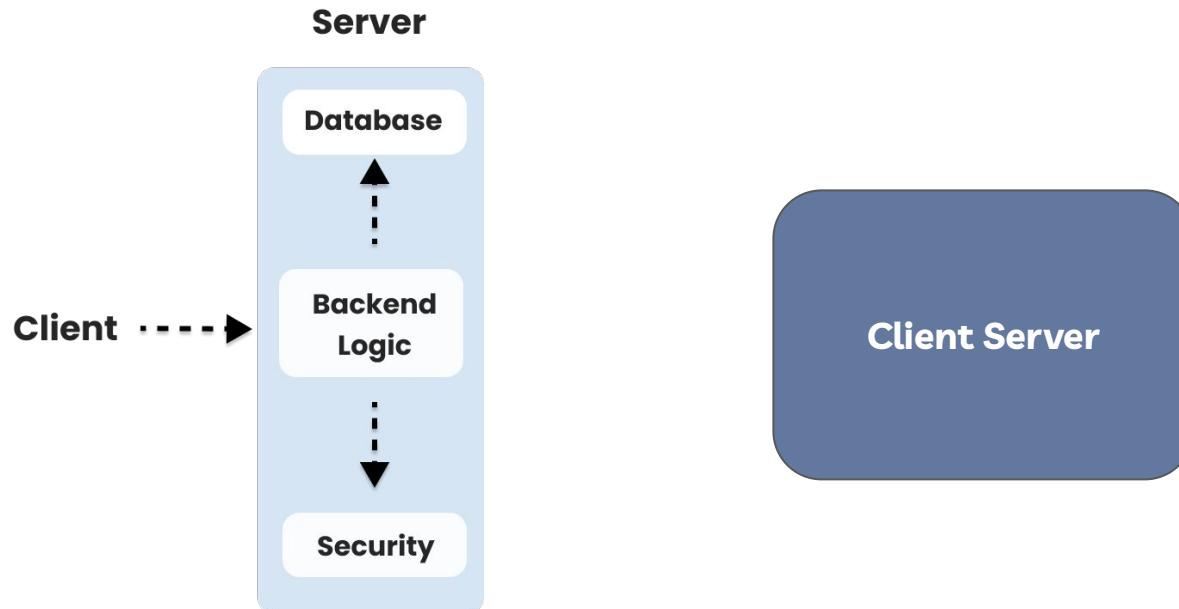


Software Architecture

Client Serverless

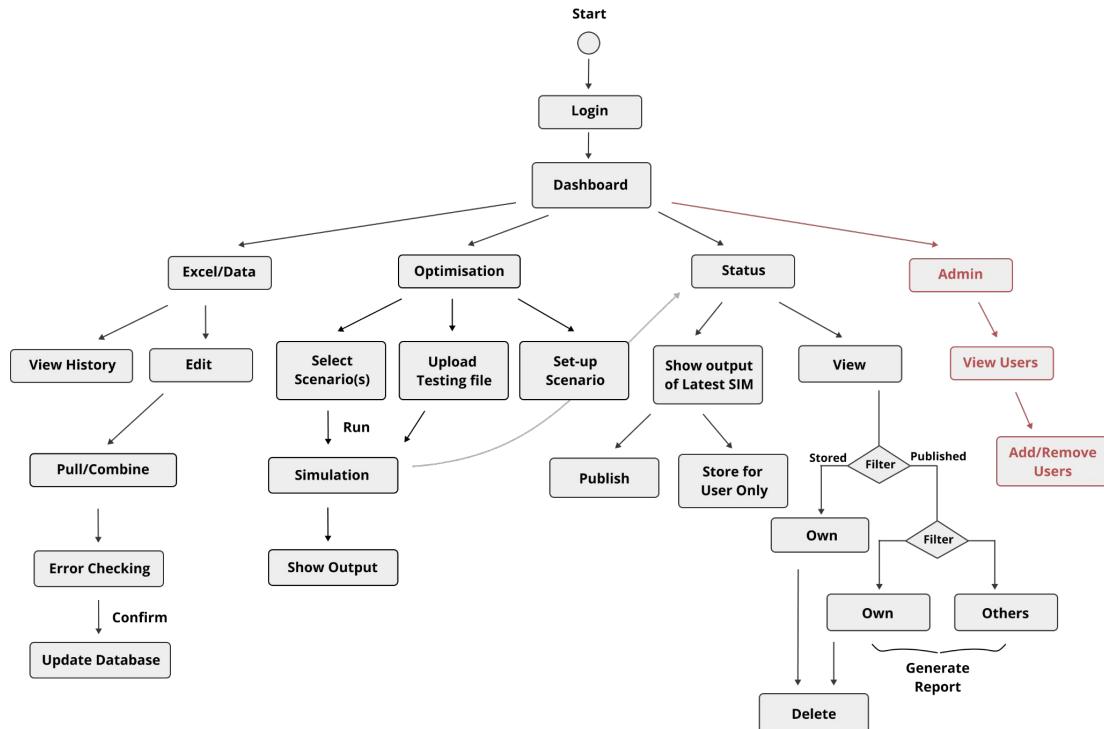
Client Server

Software Architecture



User Interface

User Interface



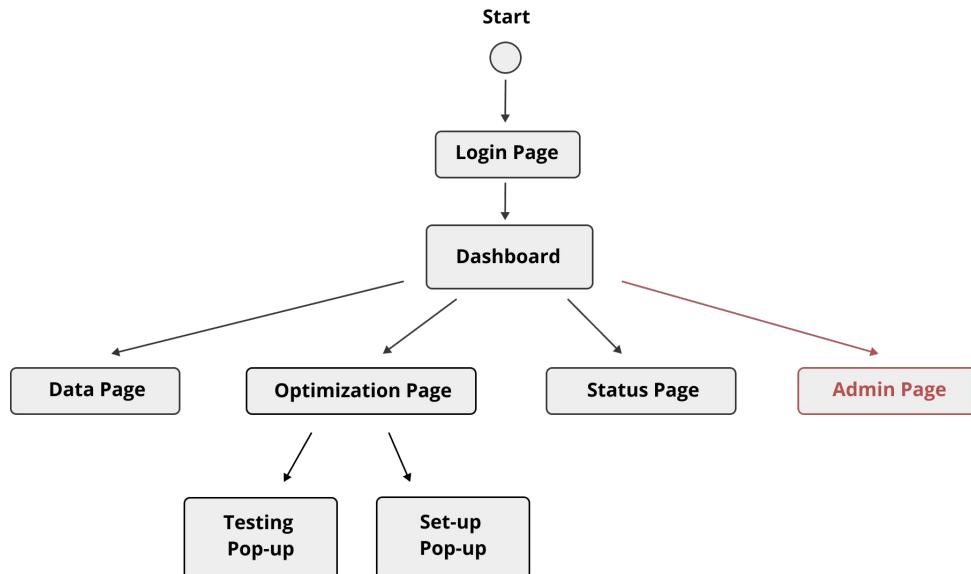
Action Flow Diagram

Contextualise user's needs

Understand user's workflow

Visualise functionalities

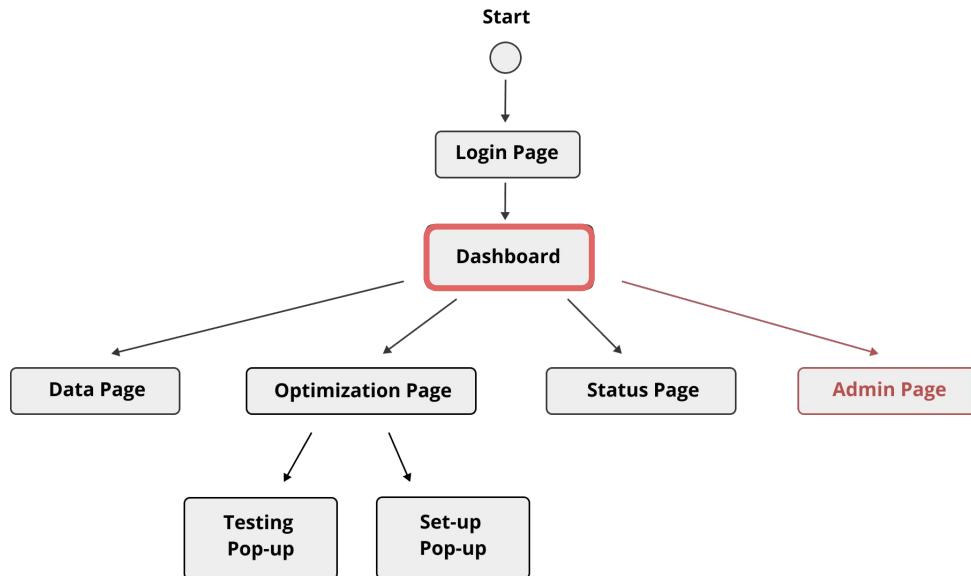
User Interface



Page Flow

Dashboard page
Data page
Optimization Page
Status Page
Admin Page (for admin user only)

User Interface



Dashboard

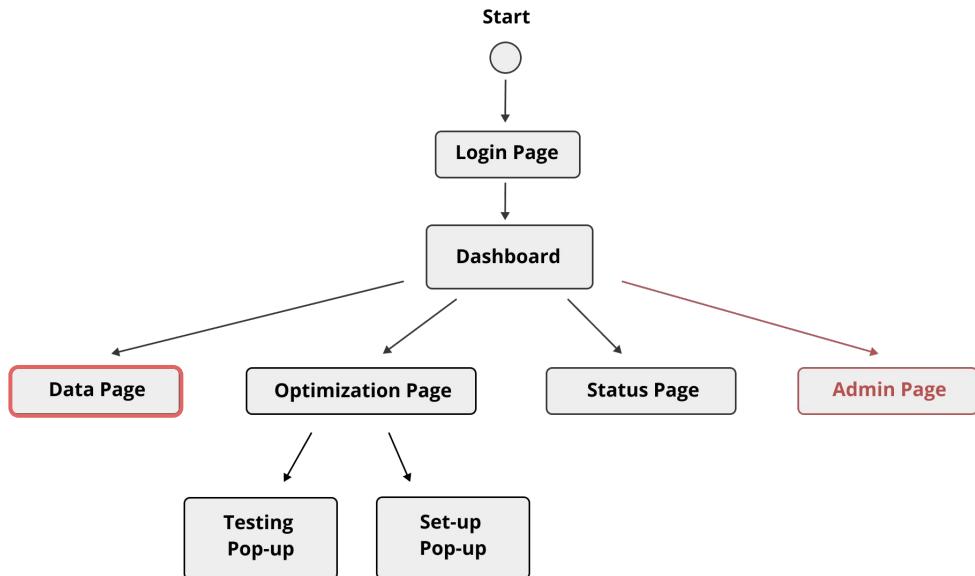
Inventory Management

Asset Management

Live Price Monitoring

Report and recommendation

User Interface

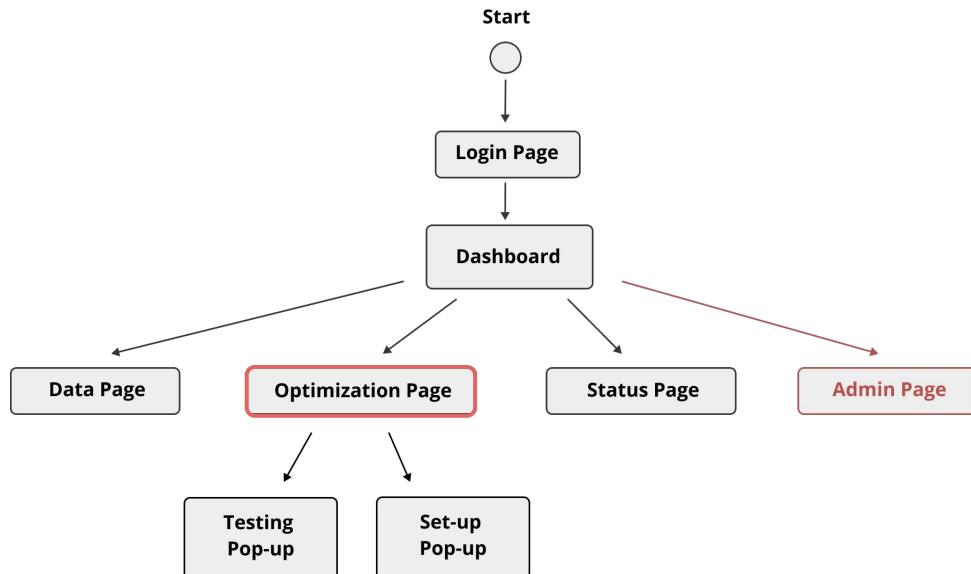


Data Page

Data Management

Update data file
View/Edit data file
Export data file

User Interface



Optimization Page

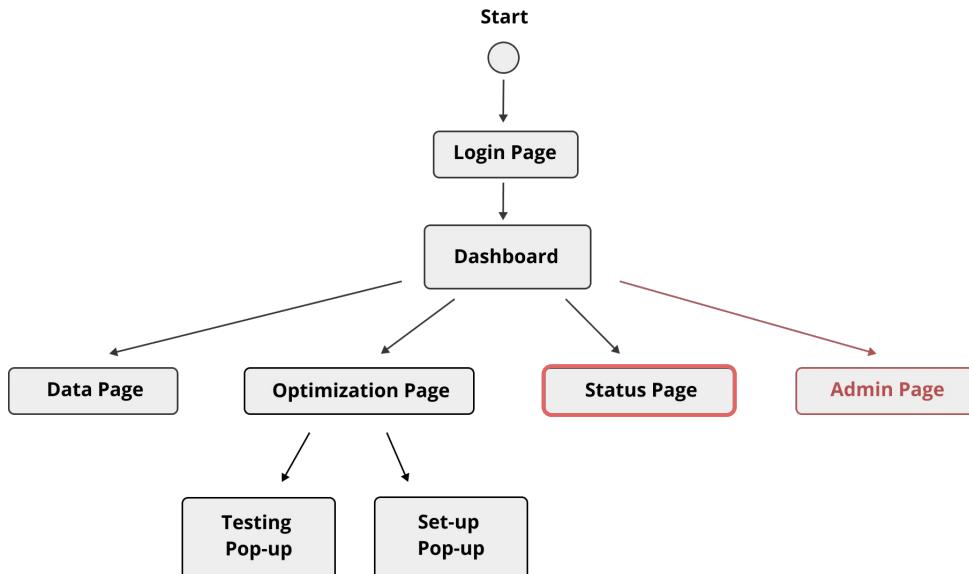
Simulation Model

Select Scenario
Set Constraints
Save Simulation output
Testing on temporary file

Supporting Features

Set new Scenarios
Adjust parameter values

User Interface



Status Page

Managing Reports

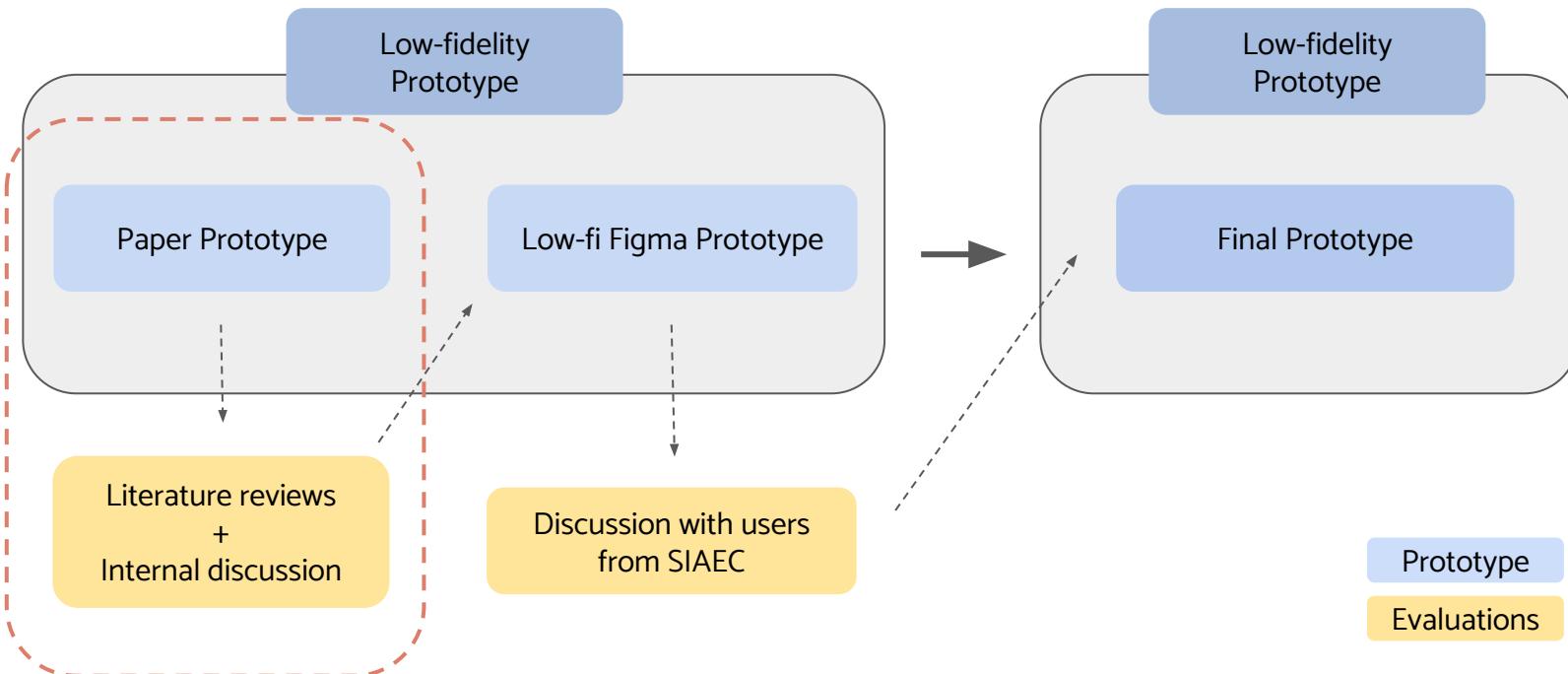
View/delete/publish report
Export report files

Sharing of Results

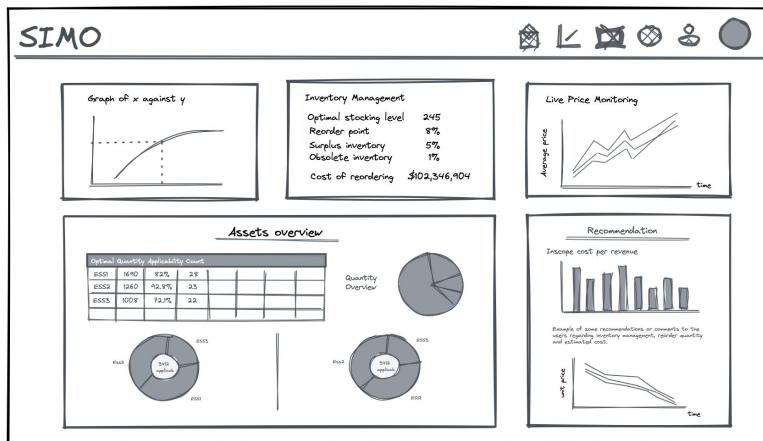
Export published report files
View published reports by other users

User Interface

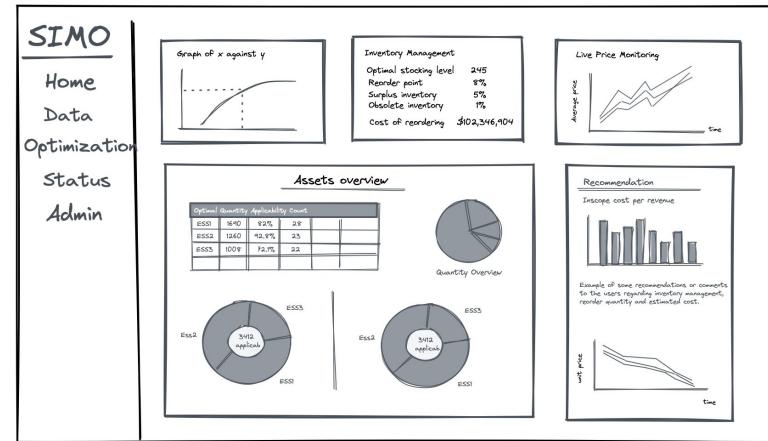
Prototypes Iterations



Navigation Bar¹³



Top Navigation Bar

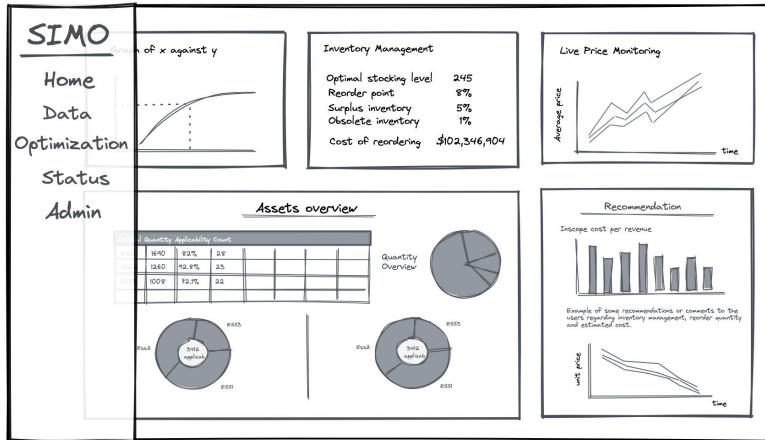


Left Navigation Bar

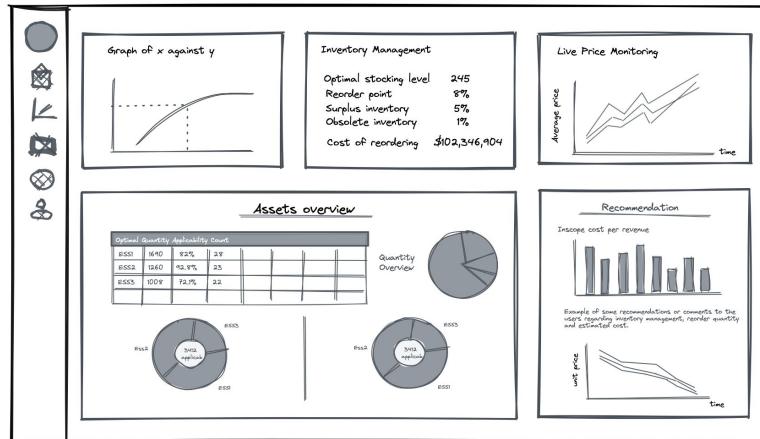
Navigation Bar

User Interface

Open



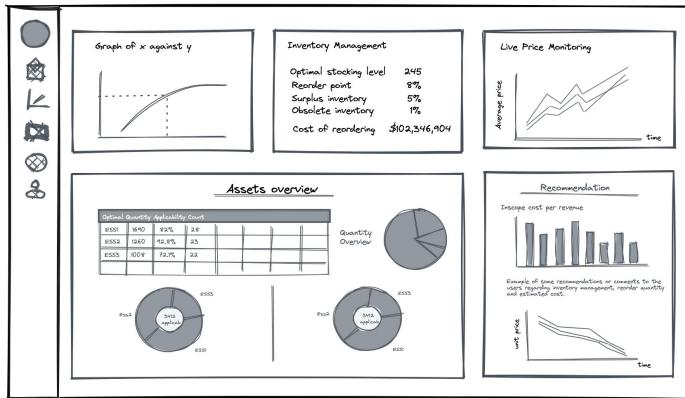
Collapsed



Collapsible Left Bar

User Interface

Improved Dashboard



Conceptual Model

Explore Various Alternative

Consolidate Screen Contents

Functionality

Navigation & Task flow

Terminology

Tech Stack

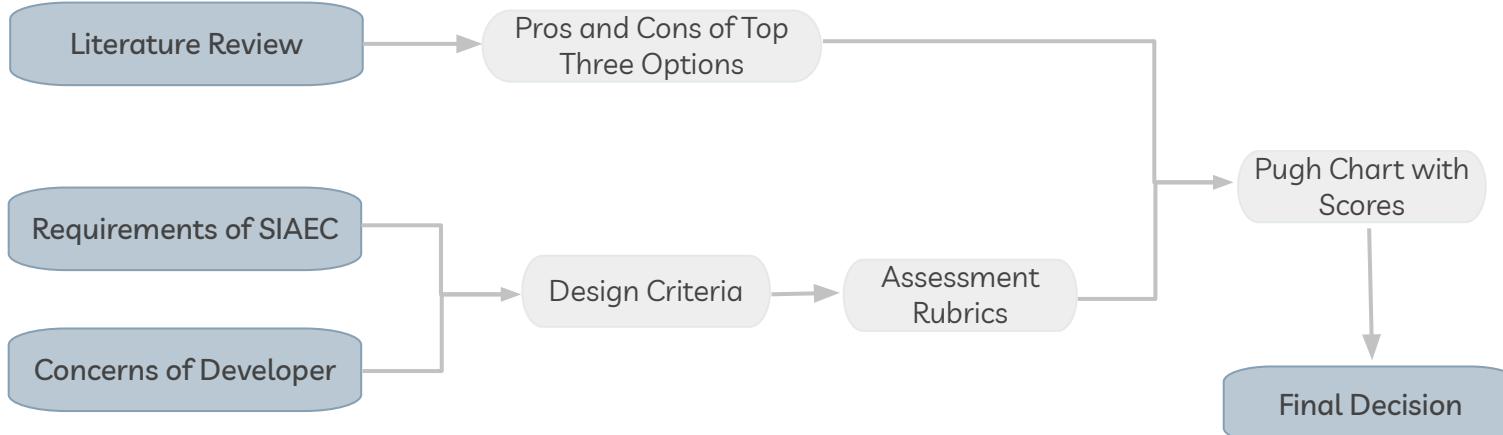
Frontend

Backend

Database Service

General Approach

For frontend, backend, and database exploration



Frontend Framework

Design Criteria	Alternative Design Concepts		
	React	Angular	Vue
Ease of Learning	+	-	-
Performance	+	-	+
Applicability of Features	+	+	0
Total	+++	-	0

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.4



Frontend Framework

	React	Angular	Vue
Ease of Learning	+	-	-
Performance	Good	Good	Good

- Familiarity to framework

Some team members have prior experience

No team members has experience

No team members has experience

Pros

Cons

Frontend Framework

	React	Angular	Vue
Performance ¹⁴	+	-	+
Pros	Only re-render when needed Support server-side rendering Virtual DOM* increases performance	MVC structure* allows separation of tasks Slower response due to the number of features	Bundle size and Tree-shaking Code splitting
Cons			

Pros

Cons

* Virtual DOM: Virtual Document Object Model;

* MVC structure: Model-View-Controller Structure

Frontend Framework

	React	Angular	Vue
Applicability of Features ¹⁴	+	+	0
Pros	Reusable components	Modular development	Template Syntax
	Unidirectional data flow	Code conventions	Track dependencies
	Changeable state	Real-time synchronization	
Cons			

Pros

Cons

Frontend Framework

Design Criteria	Alternative Design Concepts		
	React	Angular	Vue
Ease of Learning	+	-	-
Performance	+	-	+
Applicability of Features	+	+	0
Total	+++	-	0

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.4



Backend Framework

Design Criteria	Alternative Design Concepts		
	Golang Gin	Express.js	Django
Ease of Learning	-	+	0
Ease of Implementation	0	-	+
Performance	0	-	+
Total	-	-	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.5



Express



django

Backend Framework

	Golang Gin	Express.js	Django
Ease of Learning	-	+	0

- Familiarity to framework
- Available learning resources

Relative small community

Available learning resources

Available learning resources

Relative bad documentation

Some team members have experience

No team members has experience

Pros

Cons

Backend Framework

	Golang Gin ¹⁵	Express.js ¹⁶	Django ¹⁷
Ease of Implementation	0	-	+
• Code size • Flexibility	Syntax is relatively easy to learn	Same language for both frontend and backend	Readymade packages
	Easy to create middlewares	Heavy code changes due to unstable API	Model View Template
	Low reusability of code due to no OOP*	Callback functions are hard to read but needed	Lack of code conventions

Pros

Cons

Backend Framework

	Golang Gin ¹⁸	Express.js ¹⁶	Django ¹⁹
Performance	0	-	+
• Response time	Radix tree	Event-driven	MTV Architecture
• Processing speed	Built to handle concurrent tasks	Unable to handle CPU intensive task	Same language as our optimization model
	Hard to integrate with optimization model (Python)		

Pros

Cons

Backend Framework

Design Criteria	Alternative Design Concepts		
	Golang Gin	Express.js	Django
Ease of Learning	-	+	0
Ease of Implementation	0	-	+
Performance	0	-	+
Total	-	-	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.5



Express



django

Database Service

Design Criteria	Alternative Design Concepts		
	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Performance	-	+	+
Applicability of Features	-	+	0
Cost-Efficiency	-	0	0
Total	---	++	+

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.6



Database Service

	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Performance ²⁰	-	+	+
Pros	Inefficient for complex queries	Instant provisioning and auto scaling with EC2*	Sharding for cloud-base storage

- Query speed
- Concurrency capabilities

Pros
Cons

Database Service

	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Applicability of Features ²¹	-	+	0
Pros	Non-Relational	Data Replication (High Availability)	Data Replication (High Availability)
Cons		Robust Security Support	Dynamic Schemas
		Convenient integration with AWS Redshift	Limited document size
		Only vertical scaling which is costly	Non-Relational

Database Service

	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Cost-Efficiency	-	0	0

- Monthly expenditure

> SGD 300

Between SGD 100 -300

Between SGD 300 - 100

Pros

Cons

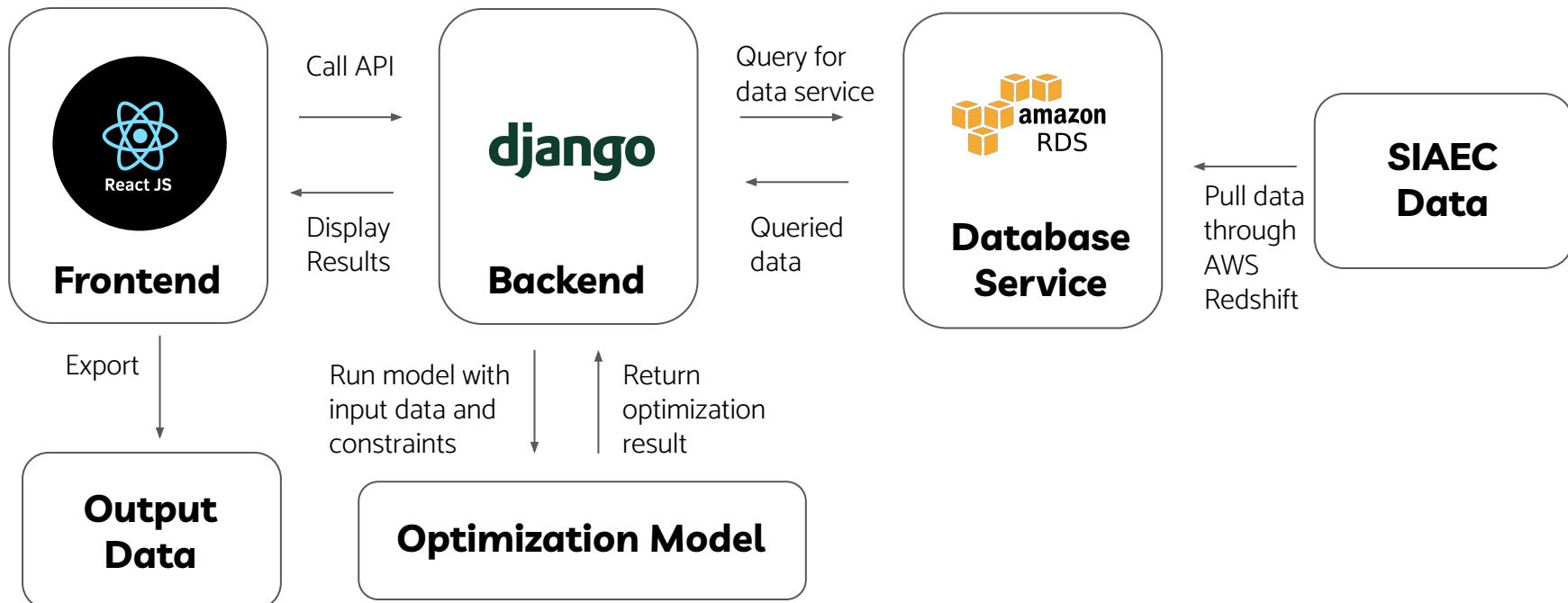
Database Service

Design Criteria	Alternative Design Concepts		
	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Performance	-	+	+
Applicability of Features	-	+	0
Cost-Efficiency	-	0	0
Total	---	++	+

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.6



Web Application Architecture Summary



Risk Assessment

Consequence Frequency Matrix
Potential Risk & Precautions
EHS

Consequence-frequency Matrix²²

		Impact		
		Small	Moderate	High
Frequency	Very High			
	High			
	Medium		Inaccurate Data Entry	Inaccurate Forecasting
	Low		Server Failure	Data Loss Security Breach

Potential Risk and Precautions

Event	Consequences	Mitigations
Inaccurate Forecasting	Potential increase in investment costs for SIAEC, leading to over provisioning	Writing test cases for algorithms
Inaccurate Data Entry	Inaccurate data displayed Decisions made based on incorrect data	Input validation
Security Breach	Stolen data Web application down	Follow the latest established security practices
Data Loss	Loss of valuable company data Web application unable to function	Data loss is hence mitigated by measures implemented by the cloud provider such as data replication
Server Failure	Web application would be down.	Will be using cloud service. Only physical disasters or virus attacks can result in a server failure



* More details can be found in Annex 1.7

Environment, Health & Safety

Safety	Sustainability
Security of Database	Ease of maintenance and update
Availability of service	Vulnerability of open source library
Confidentiality of user info	Flexibility to respond to future changes

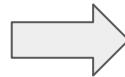
Project Management

Limitations due to Covid-19
Project Task Allocation
Project Schedule
Budget Allocation

Limitations due to COVID-19

Limitations

- Easy to fall ill
- Shortened working hours



Solution

- Clear Objectives
- Delegation of tasks

Project Task Allocation

Responsibility Matrix (Exploration and Design Phase for Prototype)									R - Responsible	S - Supporting
	Task	Justin	Olivia	Kairan	Chester	YuYing	Devid	Wenqi		
1	UI/UX design	S	R							
2	Database Schema	R		R	S		S	R		
3	Data Cleaning	R		R				R		
4	Data Integration	R		R						
5	Operations Research				S	R	S			
6	Risk Management	R			R					
7	Project Management	R			R					
8	Updating of schedule	S			R					
9	Updates to budget		R					R		
10	Report	S	S	S	S	R	R	S		
11	Slides	S	R	S	S	S	S	S		
12	Exploration for frontend	S	R							
13	Exploration for database	R		S						
14	Exploration for backend	S		R						
15	Exploration for system architecture	S		R						

Project Schedule

March		April				May			
Concept and application selection									
	Wireframe for UI								
	Research theory and application								
	Formulation and completion of Just In Time Provisioning								
		Database Planning							
		Frontend and Backend Exploration							
			Review 2 Report and Slides						
			Just in Time Provisioning						
					Asset Management				
									Frontend and Backend Implementation, Testing
									Price Monitoring System

Project Schedule

June				July				August
Frontend and Backend Implementation Testing								
Price Monitoring System								
Review 3 slides								
Model Fine Tuning								
				Deploy Frontend and Backend				
					User Testing			
				Final Report and software documentation				
						Poster and summary of analysis result		
								Exhibition and Final Presentation

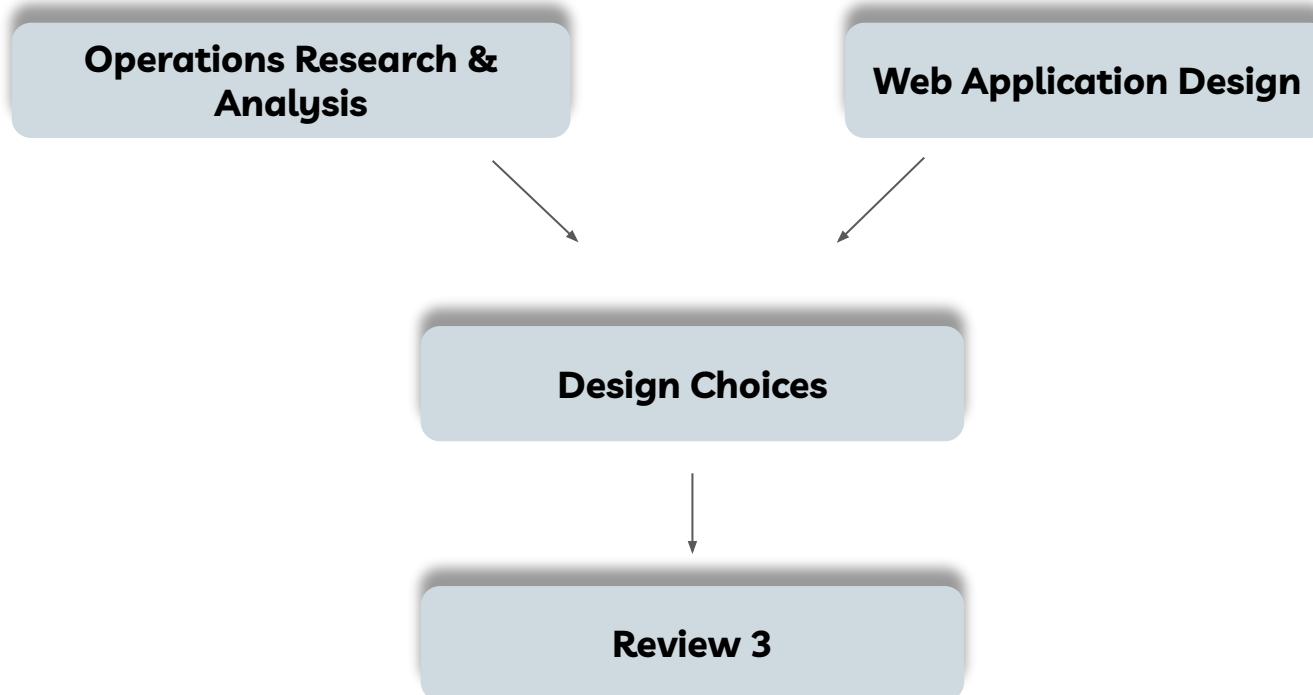
Expected Expenditure (SGD)	May	Jun	Jul	Aug
Tools				
Jira	52.50	52.50	52.50	52.50
Softwares				
RDS	200	200	200	200
EC2	287.50	287.50	287.50	287.50
Cloudfront	0	0	0	10
SSL Cert	0	0	0	25
Transportation				
Taxi	100	100	100	100
Total Expenditure	590	590	590	625

Budget Allocation

	May	Jun	Jul	Aug
Total Fund Available (SGD)	4,000	3,410	2,820	2,230
Total Expenditure (SGD)	590	590	590	625
Total Fund Remaining (SGD)	3,410	2,820	2,230	1,605

Conclusion

Conclusion



Q&A

Reference

References

Content

1. Our company profile. SIA Engineering Company. (n.d.). Retrieved March 2, 2022, from https://www.siaec.com.sg/company_profile.html
2. Ohri, A. (2017). 10 Popular Regression Algorithms In Machine Learning Of 2022. Retrieved 21 April 2022, from <https://www.jigsawacademy.com/popular-regression-algorithms-ml/#:~:text=Regression%20is%20a%20type%20of,the%20most%20popular%20regression%20algorithms>.
3. Wilkinson, G. (2020). Asset management optimization for repairable spare parts. Retrieved 21 April 2022, from <https://www.anylogic.com/blog/asset-management-optimization-for-repairable-spare-parts/>
4. Lye, K., & Chan, L. (2007). A Virtual Warehouse Simulation Tool for Aerospace Rotables Management. IEEE Aerospace Conference Proceedings.
5. Demand classification: Why forecastability matters - frepple APS. frePPLe. (2021, February 16). Retrieved April 12, 2022, from <https://frepple.com/blog/demand-classification/>
6. Chawla, G., & Miceli, V. (1970, January 1). Demand forecasting and inventory management for Spare Parts. Demand Forecasting and Inventory Management for Spare Parts. Retrieved April 20, 2022, from <https://dspace.mit.edu/handle/1721.1/121291>
7. Syntetos, A. A., Babai, M. Z., Lengu, D., & Altay, N. (1970, January 1). Distributional assumptions for parametric forecasting of intermittent demand. SpringerLink. Retrieved April 21, 2022, from https://link.springer.com/chapter/10.1007/978-0-85729-039-7_2#citeas
8. Pinçé, Ç., Turrini, L., & Meissner, J. (2021, July 3). Intermittent demand forecasting for spare parts: A critical review. Omega. Retrieved March 27, 2022, from <https://www.sciencedirect.com/science/article/pii/S0305048321001225>
9. Ofiwe, M. (2021, October 28). Metasearch Engines. SEMrush. Retrieved April 21, 2022, from <https://www.semrush.com/blog/metasearch-engine/>

References

Content

10. Wattanarat, V., Phimphavong, P., & Matsumaru, M. (2011). Demand and Price Forecasting Models for Strategic and Planning Decisions in a Supply Chain.
11. Why use serverless computing? | Pros and cons of serverless. (n.d.). Cloudflare. Retrieved April 22, 2022, from <https://www.cloudflare.com/en-gb/learning/serverless/why-use-serverless/>
12. Lange, K. (2021, February 25). What's Serverless? Pros, Cons & How Serverless Computing Works. BMC Software. Retrieved April 22, 2022, from <https://www.bmc.com/blogs/serverless-computing/>
13. Bakusevych, T. (2021, March 2). *Top vs side navigation: Which one is better for your product?* UX Collective. Retrieved April 22, 2022, from <https://uxdesign.cc/top-navigation-vs-side-navigation-wich-one-is-better-24aa5d835643>
14. Borrelli, P. (2021, October 26). Angular vs. React vs. Vue.js: Comparing performance. LogRocket Blog. Retrieved April 21, 2022, from <https://blog.logrocket.com/angular-vs-react-vs-vue-js-comparing-performance/>
15. Jonah, V. (2021). 5 top Go web frameworks - LogRocket Blog. Retrieved 21 April 2022, from <https://blog.logrocket.com/5-top-go-web-frameworks/>
16. Volodymyr, T. (2017). Express.js Mobile App Development: pros and cons of Node.js framework. Retrieved 21 April 2022, from <https://apiko.com/blog/express-mobile-app-development/>
17. Bhatt, S. (2020). Pros and Cons of Django Framework for App Development - DZone Web Dev. Retrieved 21 April 2022, from <https://dzone.com/articles/pros-and-cons-of-django-framework-for-app-developm>
18. Gin Web Framework. Retrieved 21 April 2022, from <https://gin-gonic.com/>
19. DataFlair,. Retrieved 21 April 2022, from <https://data-flair.training/blogs/django-advantages-and-disadvantages/>
20. Wodehouse, C. (2018). Firebase vs. AWS vs. MongoDB: 3 Technologies Behind Modern Software Stacks. Retrieved 21 April 2022, from <https://www.business2community.com/brandviews/upwork/firebase-vs-aws-vs-mongodb-3-technologies-behind-modern-software-stacks-02147649>

References

Content

21. Singh, M. (2021). Social Network for Programmers and Developers. Retrieved 21 April 2022, from <https://morioh.com/p/fd327efa3b32>
22. van Westen, C. (n.d.). 5.5 Risk assessment methods. Caribbean Disaster Emergency Management Agency - CDEMA. Retrieved April 20, 2022, from
<https://www.cdema.org/virtuallibrary/index.php/charim-hbook/methodology/5-risk-assessment/5-5-risk-assessment-methods>

References

Design Credit

Presentation template by [Slidesgo](#)

Icons by [Flaticon](#)

Infographics by [Freepik](#)

Images

<https://www.simplilearn.com/data-analysis-methods-process-types-article>

https://www.123rf.com/photo_36638346_modern-notebook-computer-with-future-technology-media-symbols.html

Annex

Short-term Provisioning

Criteria\Score	-	0	+
Data Requirement	The method requires a few years of historical data	The method requires a few months a historical data	The method requires a month worth of historical data
Accuracy	The method produces poor accuracy result	The method produces acceptable accuracy result	The method produces exceptional accuracy result
Time	The method requires a lot of time to implement	The method uses moderate amount of time to implement	The method requires very little time to implement
Ease of implementation	The method is computationally intensive with many mathematical equations and concepts	The method is moderately easy to implement with fair amount of mathematical equations and concepts	The method is easy to implement with little mathematical equations and concepts

Annex 1.2

Price Monitoring System (Price Acquisition)

Criteria\Score	-	0	+
Affordability	Software costs more than SGD 100	Software costs less than SGD 100	This software is free to use
Ease of Integration	Software is difficult to integrate relative to the other alternatives	Software is reasonably difficult to integrate relative to the other alternatives	Software is easy to integrate relative to the other alternatives
Adaptability	Requires major tuning whenever website is changed relative to the other alternatives	Requires moderate tuning whenever website is changed relative to the other alternatives	Requires little to no tuning whenever website is changed relative to the other alternatives

Price Monitoring System (Price Forecasting)

Criteria\Score	-	0	+
Data Requirement	The method requires a few years of historical data	The method requires a few months a historical data	The method requires a month worth of historical data
Accuracy	The method produces poor accuracy result	The method produces acceptable accuracy result	The method produces exceptional accuracy result
Time	The method requires a lot of time to implement	The method uses moderate amount of time to implement	The method requires very little time to implement
Ease of implementation	The method is computationally intensive with many mathematical equations and concepts	The method is moderately easy to implement with fair amount of mathematical equations and concepts	The method is easy to implement with little mathematical equations and concepts

Annex 1.4

Frontend Framework

Criteria\Score	-	0	+
Ease of Learning	All team members have no prior knowledge with the framework	Some team members have some prior knowledge with the framework	Every members have prior experience with the framework
Performance	Has slow rendering speed, slower response speed with increased number of features relative to the other alternatives	Has a moderate rendering speed, no significant reduction in response speed with increased number of features relative to the other alternatives	Has fast rendering speed, no significant reduction in response speed with increased number of features relative to the other alternatives
Applicability of Features	Offers few relevant features fulfilling the needs and constraint of the project relative to the other alternatives	Offers some relevant features fulfilling the needs and constraint of the project relative to the other alternatives	Offers many relevant and useful features fulfilling the needs and constraint of the project relative to the other alternatives

Annex 1.5

Backend Framework

Criteria\Score	-	0	+
Ease of Learning	All team members have no experience with the framework, few learning resources available	All team members have no experience with the framework, some learning resources available	Some team members have experience with the framework, some learning resources available
Ease of Implementation	Framework requires a lot of code, rigid in implementation	Framework requires a moderate amount of code, relatively flexible in implementation	Framework requires a moderate amount of code, flexible in implementation
Performance	Slow response time relative to the other alternatives	Moderate response time relative to the other alternatives	Fast response time relative to the other alternatives

Annex 1.6

Database Service

Criteria\Score	-	0	+
Performance	Slow reads and writes with poor concurrency capabilities relative to the other alternatives	Fast reads and writes with moderate concurrency capabilities relative to the other alternatives	Fast reads and writes with good concurrency capabilities relative to the other alternatives
Applicability of Features	Has few relevant features offered by service provider relative to the other alternatives	Has some relevant features offered by service provider relative to the other alternatives	Has many relevant and useful features offered by service provider relative to the other alternatives
Cost-Efficiency	Monthly expenditure is above SGD 300	Monthly expenditure is between SGD 300 and SGD 100	Monthly expenditure is below SGD 100

Potential Risk and Precautions

Event	Description	Consequences	Mitigations
Inaccurate Forecasting	Caused by the algorithm's inaccuracy to forecast out the optimal number of parts for SIAEC to provision or to keep as active stock	Potential increase in investment costs for SIAEC, leading to over provisioning	Writing test cases for algorithms
Inaccurate Data Entry	Data input into the database is wrong	Inaccurate data displayed Decisions made based on incorrect data	Input validation
Security Breach	Caused by human error and loopholes in software.	Stolen data Web application down	Follow the latest established security practices
Data Loss	Caused by human error, viruses, natural disasters, power failure	Loss of valuable company data Web application unable to function	Data loss is hence mitigated by measures implemented by the cloud provider such as data replication
Server Failure	Caused by disk failure, virus attack, failed updates and physical disasters	Web application would be down.	Will be using cloud service. Only physical disasters or virus attacks can result in a server failure

OUR TEAM



Chester
Lim Zi Hao



Cui Wenqi



Olivia
Chen Ken



Justin
Peng Zheng
Wei



Chia Yu Ying



Hour
Youlin Sereydevid



Sun Kairan

TABLE OF CONTENTS

01

OVERVIEW

Company Background, Problem Framing, Problem Analysis, Milestones, User Needs & Constraints, Refined Problem Statement

02

DESIGN DIRECTION

Concept Generation, Optimizations Research & Analysis, Web Application Design

03

RISK ASSESSMENT

Consequence Frequency Matrix, Potential Risk & Precautions

04

PROJECT MANAGEMENT

Project Task Allocation, Project Schedule, Budget Allocation

05

CONCLUSION

Overview

Company Background

Problem Framing

Problem Analysis

Milestones

User Needs & Constraints

Refined Problem Statement

Company Background¹

- Singapore Airlines Engineering Company (SIAEC) is Asia's foremost maintenance, repair and overhaul (MRO) provider
- Delivers integrated solutions to large client bases consisting of both international airline and aerospace companies.
- Provides frontline maintenance services to more than 60 airlines* that fly through Singapore whilst ensuring a high level of punctuality for their customers' flight takeoffs.

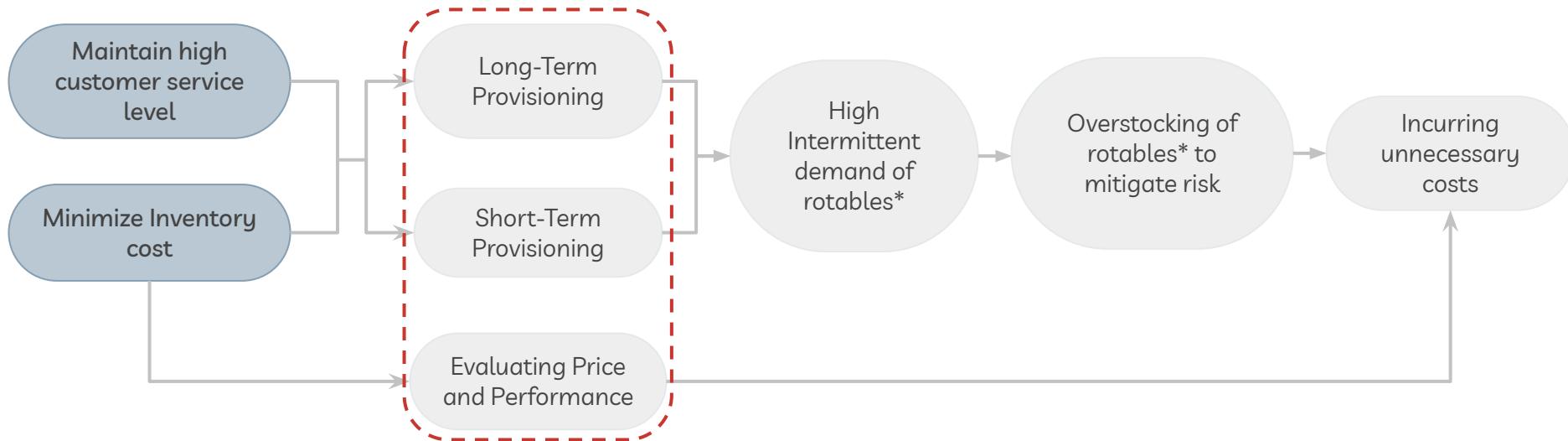
*Some companies include Singapore Airlines, SilkAir and TigerAir

Problem Framing

Despite efforts by SIAEC in deploying inventory management practices, the company ***still overprovision rotables***, incurring ***unnecessary capital***. The underlying causes of this overprovisioning include ***inefficient inventory management practices, forecasting techniques, and inflexible interface for managing inventory.***

Problem Analysis

Goals for SIAEC

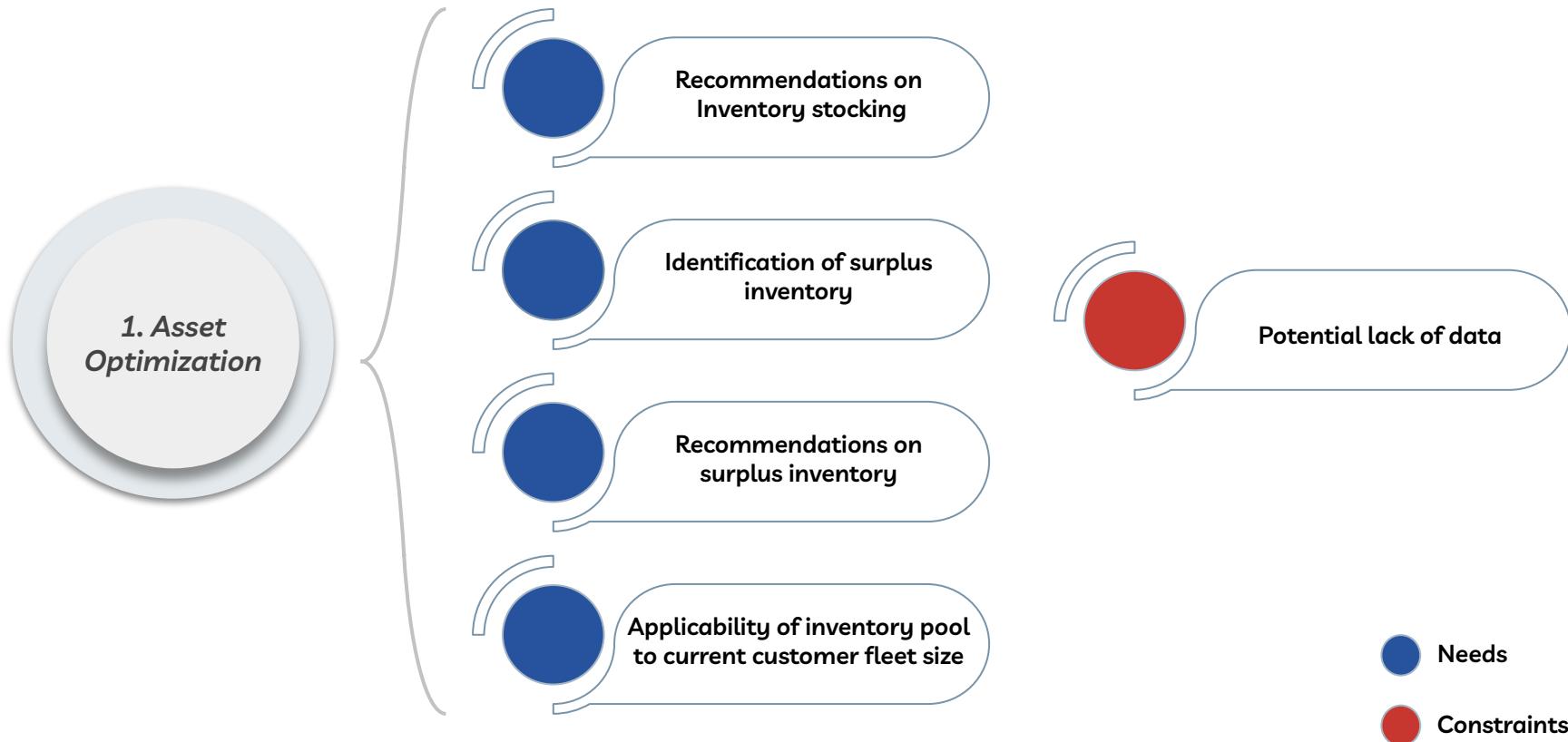


*Rotables are aircraft components which are removed, replace or inspected at intervals and they consist a significant percentage in a typical commercial aircraft.

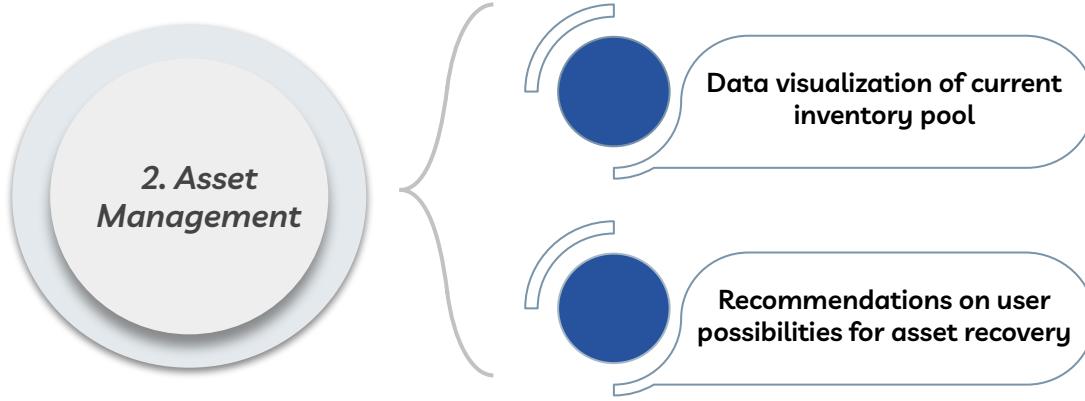
Milestone



User Needs & Constraints

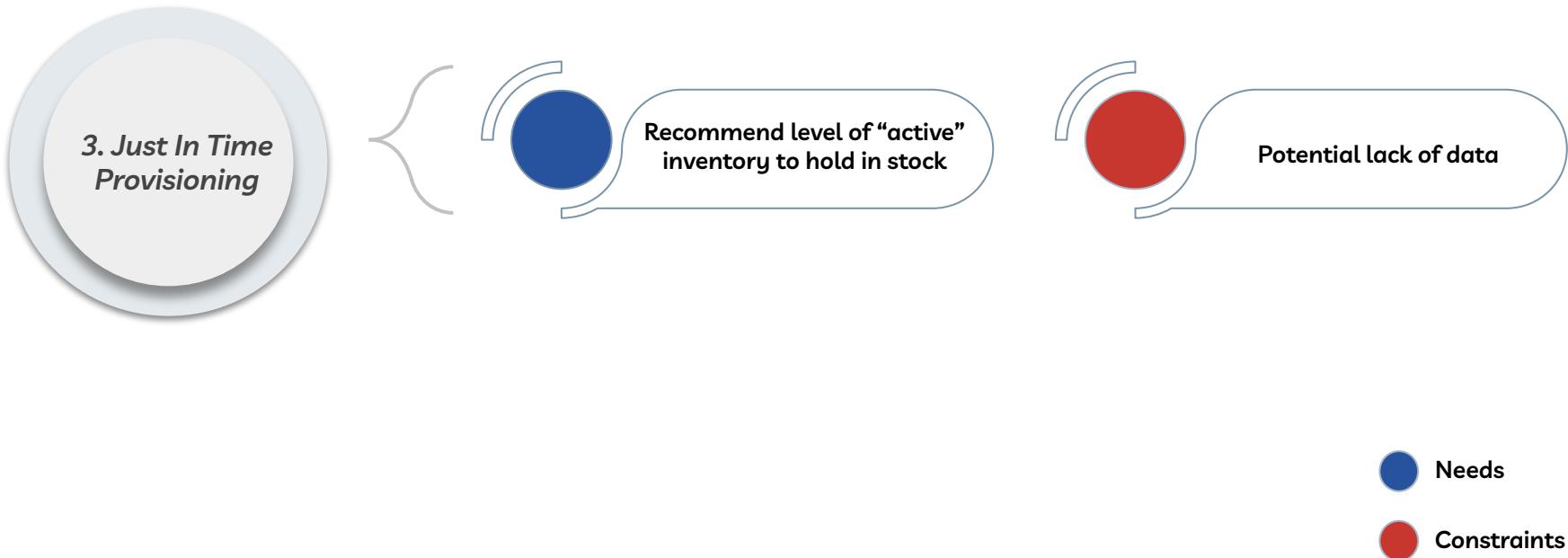


User Needs & Constraints



- Needs
- Constraints

User Needs & Constraints



User Needs & Constraints



- Needs
- Constraints

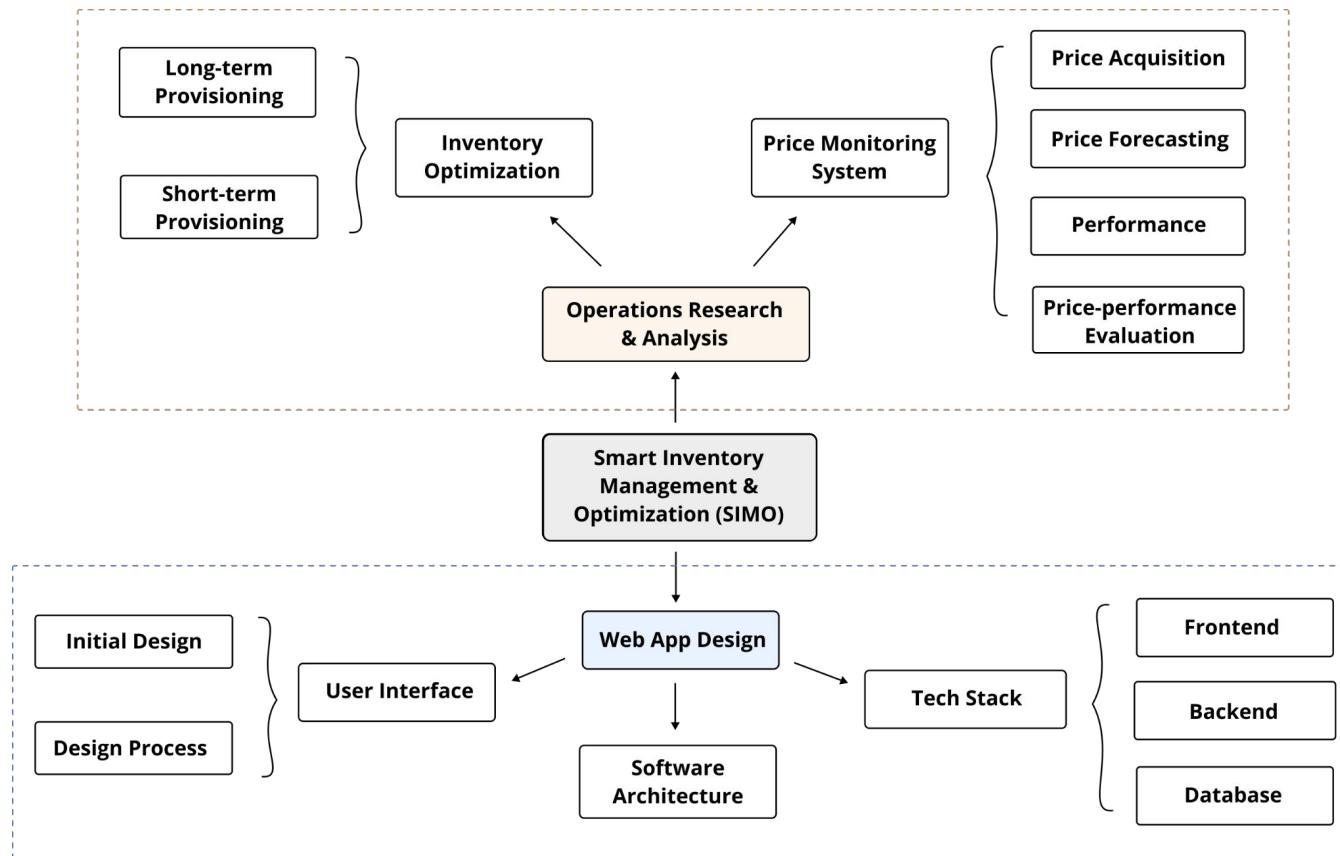
Refined Problem Statement

This project aims to **create a web-based application** with dynamic features such as having **variable inputs, scenario-based optimization** and **data visualization** to help manage and optimize rotables in hopes of a more **user-friendly and cost-saving alternative** for the company to fit into their current workflow.

Design Direction

Concept Generation
Optimizations Research & Analysis
Web Application Design

Concept Generation



Operations Research & Analysis

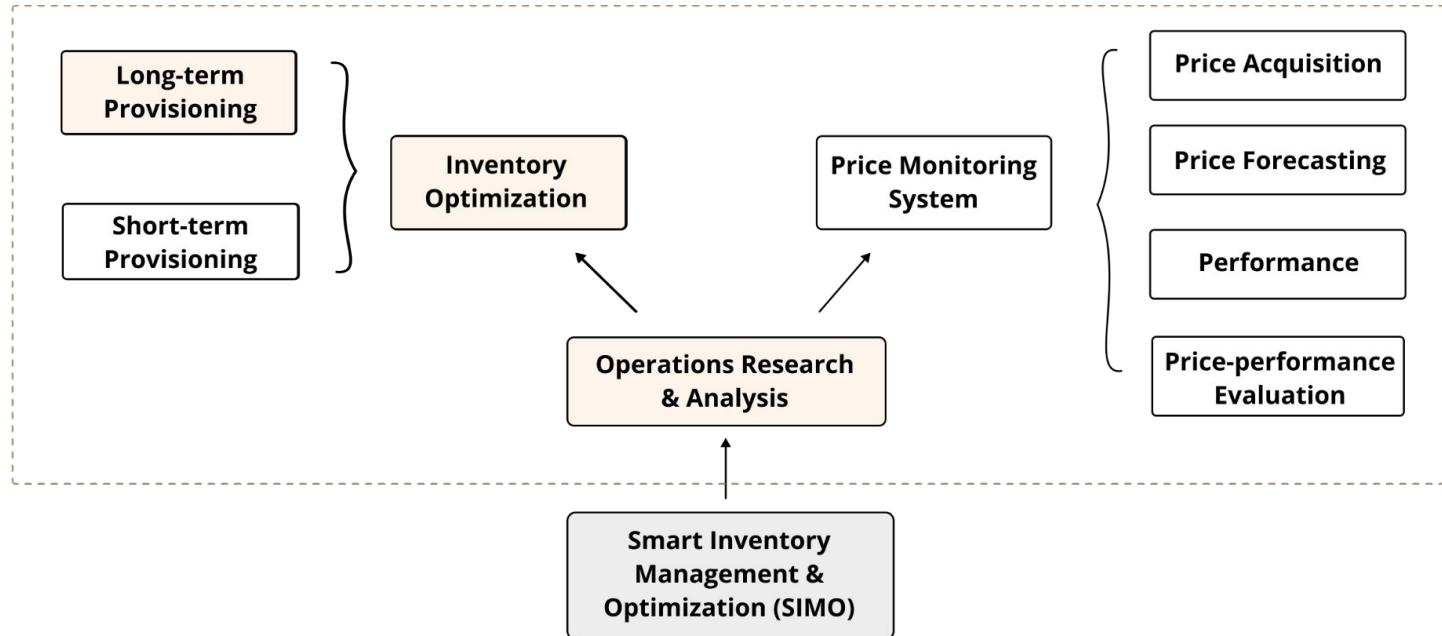
Inventory Optimization

Long-term Provisioning
Short-term Provisioning

Price Monitoring System

Price Acquisition
Performance
Price-performance Evaluation

Long-term Provisioning



Milestone

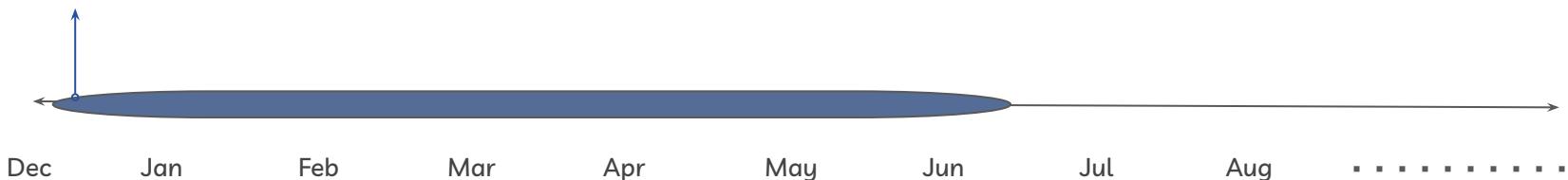


Milestone (Long-term Provisioning)



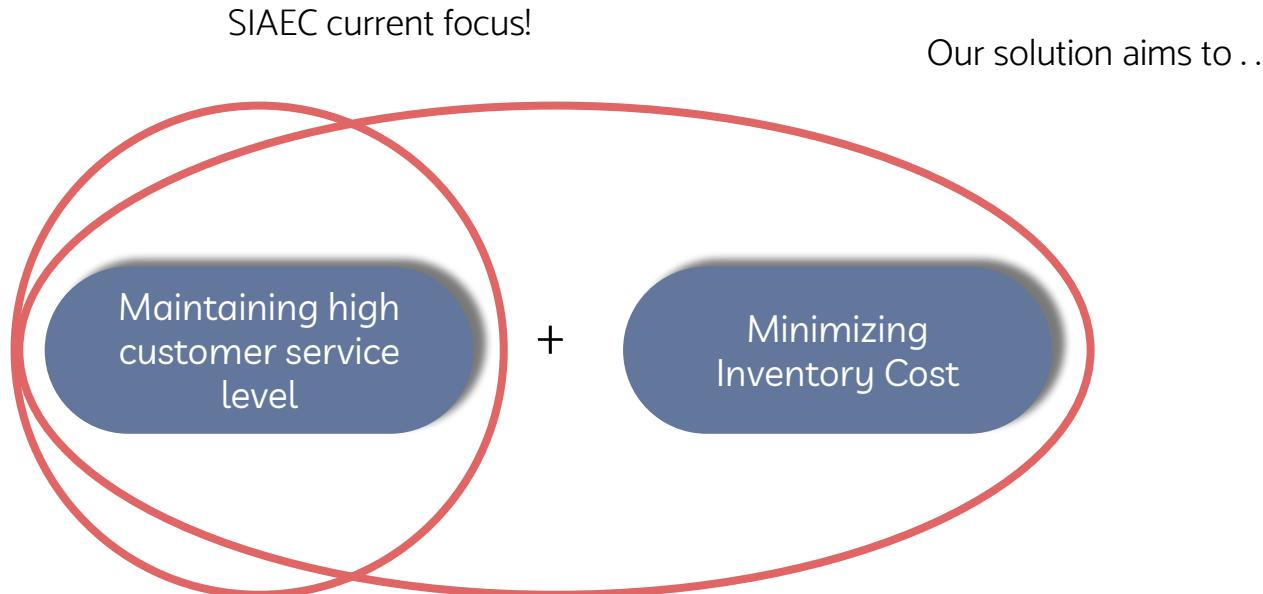
Long-Term Provisioning

Determine Number of Quantity to be topped up 6 months in advance for each type of rotable



Long-Term Provisioning

GOALS:





Approach A

Optimization model

Long-term Provisioning

Approach B

Improving Accuracy & Quality of
Input Data



Approach C

Statistical Distribution of Demand



Approach A

Optimization model

Long-term Provisioning

Approach B

Improving Accuracy & Quality of
Input Data

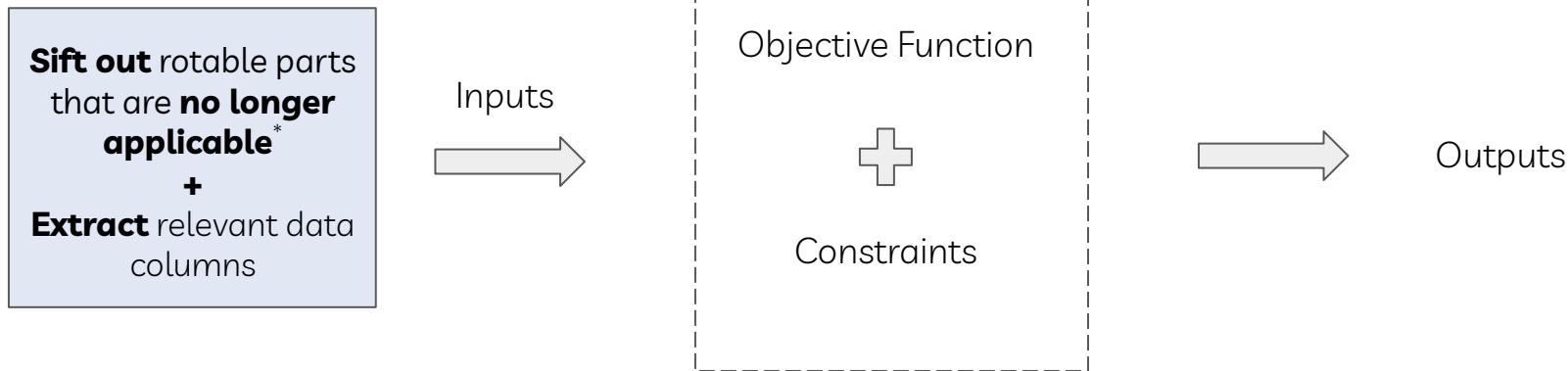


Approach C

Statistical Distribution of Demand

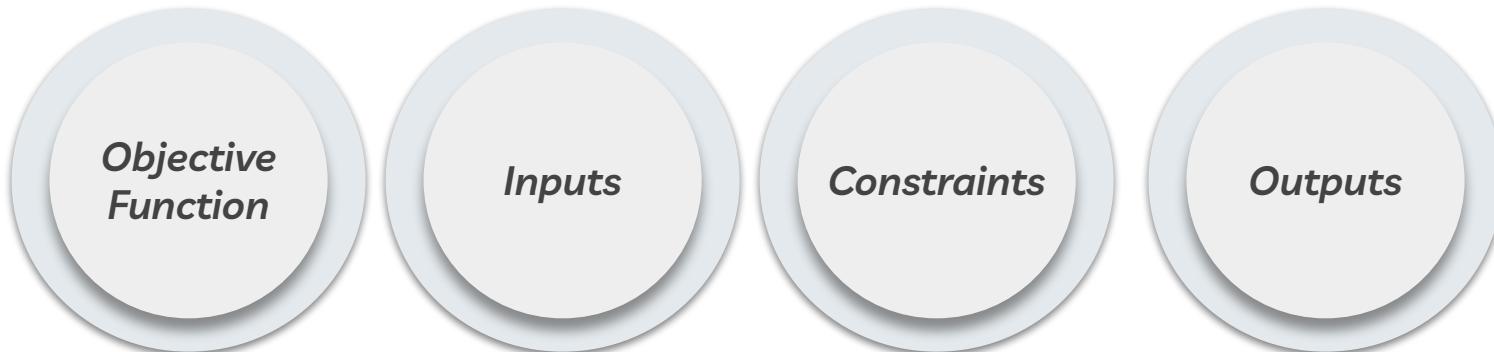
Long-term Provisioning

Optimization Model



* Applicable: Non-applicable rotatable parts are parts that are no longer part of the contractual agreement between SIAEC and their customers hence, SIAEC will not need to provide for these parts to customer

Optimization Model (Formulation)

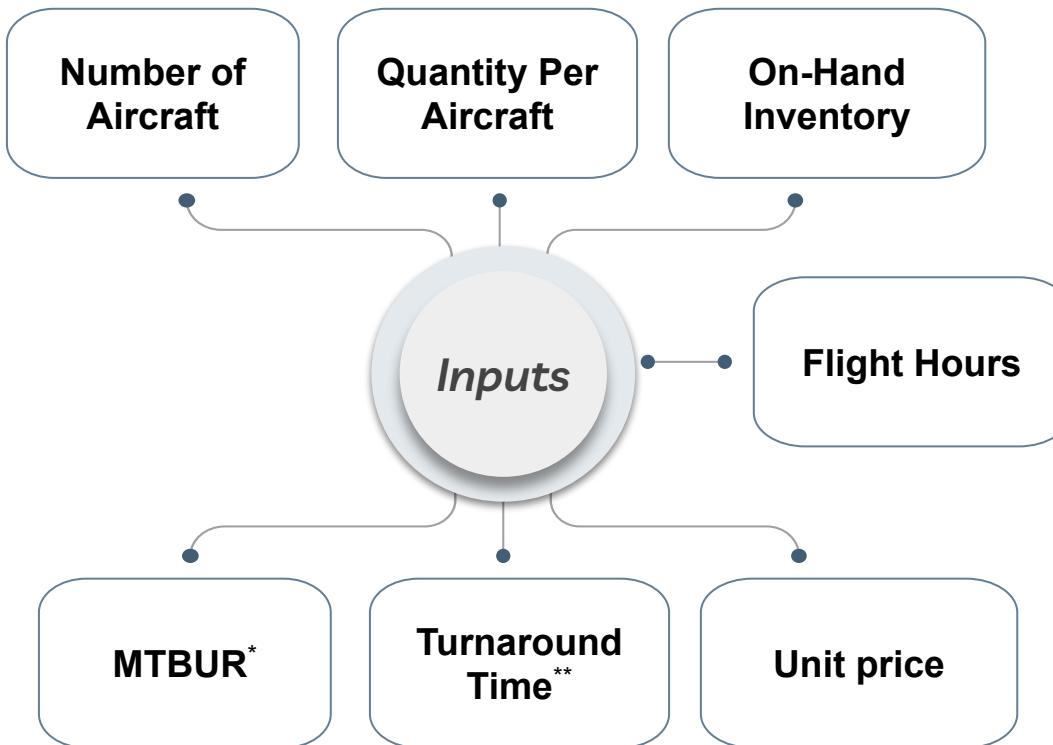


Optimization Model (Objective Function)



- **Minimize** the **total cost** to purchase the required rotables

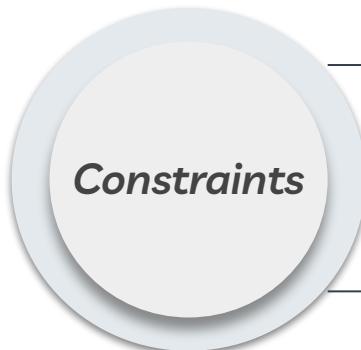
Optimization Model (Inputs)



*MTBUR: Mean Time Before Unscheduled Removal;

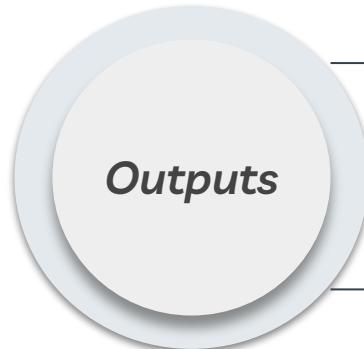
**Turnaround Time includes the repair time and logistics/transportation time

Optimization Model (Constraints)



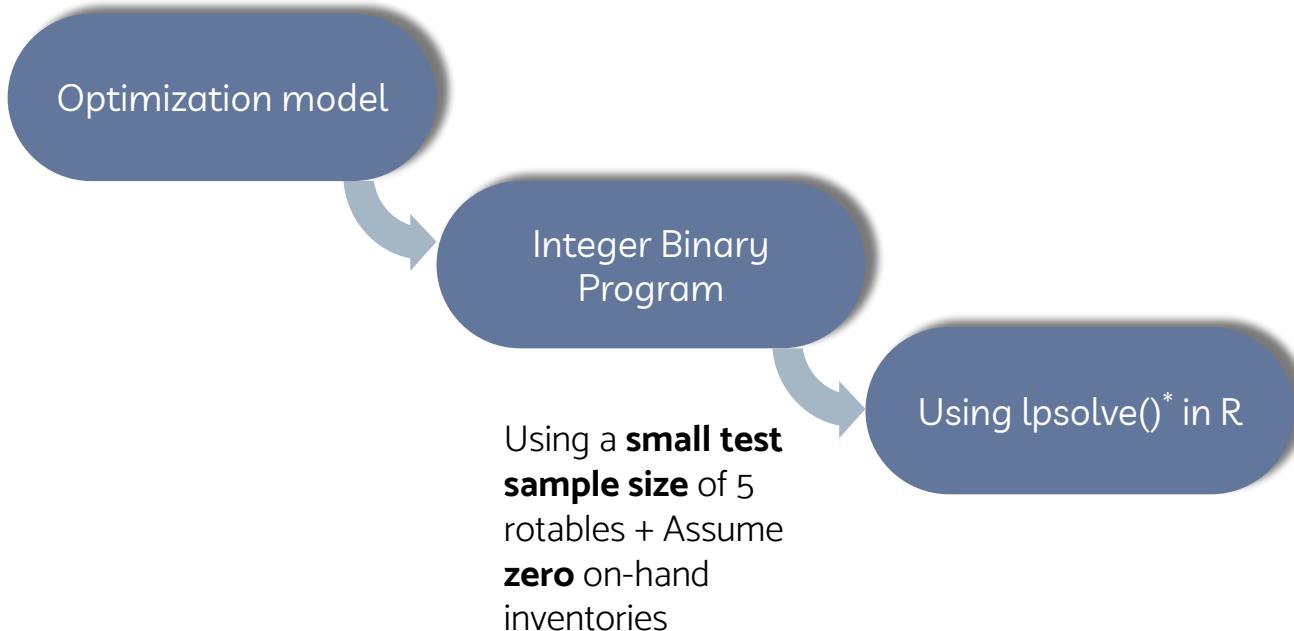
- **Service level** for each class* of rotables are **to be met**
- **Total cost** Incurred should **not exceed** SIAEC budget
- A certain number of specific types of rotables are **always to be available** in the warehouse

Optimization Model (Outputs)



- **Top Up** Quantity for each type of rotable
- Acquired **Service Level** for each class of rotables
- **Total cost** incurred

Optimization Model (Experiment)



Optimization Model (Experiment* result)

Class**	Part Number	Unit Price (\$)	Quantity to hold (SIAEC's method)	Quantity to hold (Our optimization model)
2	12-34	17,712	1	1
2	12-35	17,712	0	0
2	12-36	9501	1	0
2	12-37	4678	1	1
2	12-38	83,971	2	1

SIAEC's current method

Average Service Level: 98.22% (>95%)

Total Investment: USD 199,836

~50%
reduction

Our Optimization model

Average Service Level: 95.82% (>95%)

Total Investment: USD 106,363

*Experiment: To simplify our model, we have assumed on-hand inventory to be zero for this experiment

**Class: Class 2 rotables need to achieve on average a service level of 95%



Approach A

Optimization model

Long-term Provisioning

Approach B

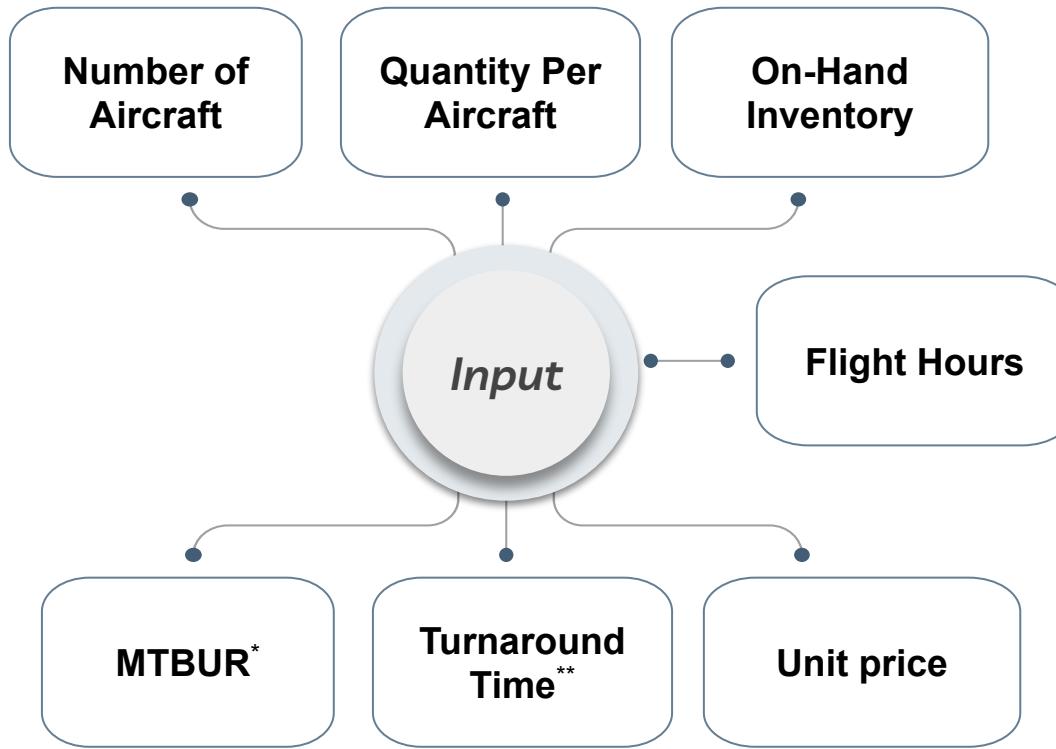
Improving Accuracy & Quality of
Input Data



Approach C

Statistical Distribution of Demand

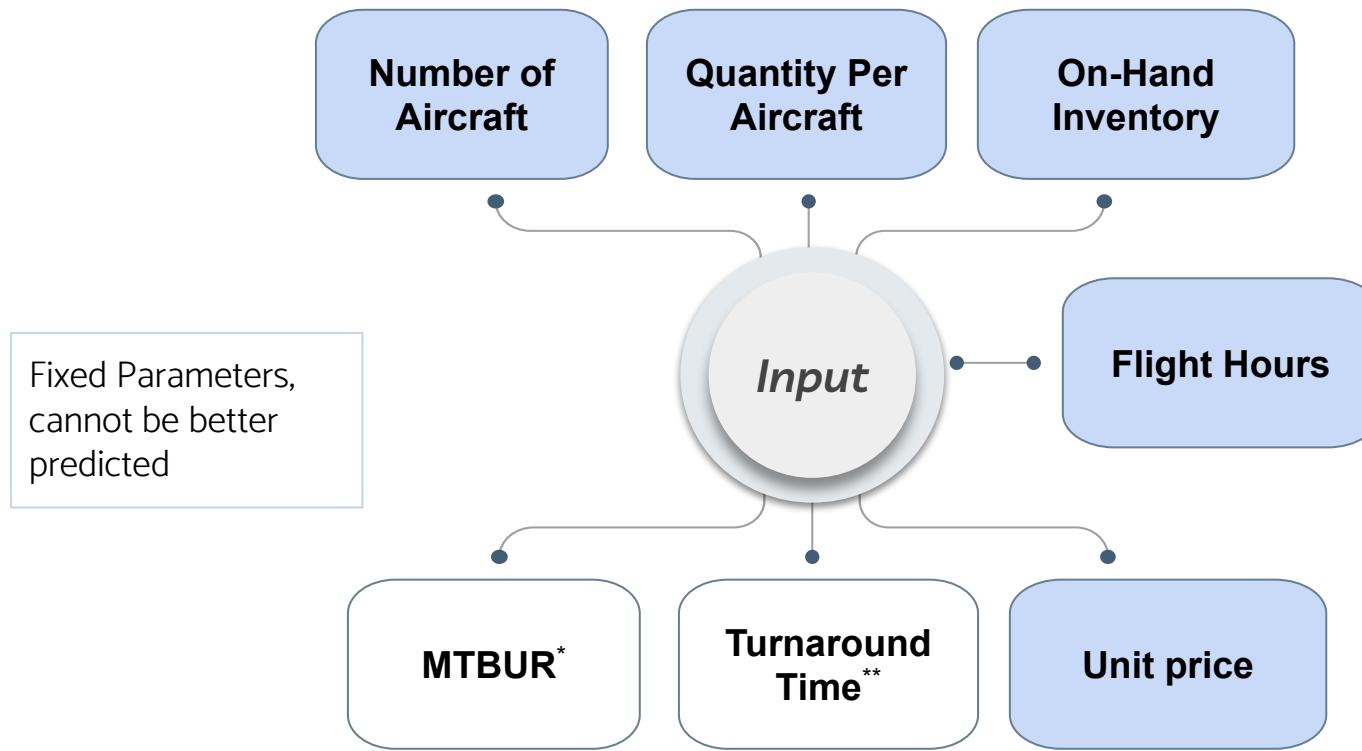
Improving Quality of Input Data



*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

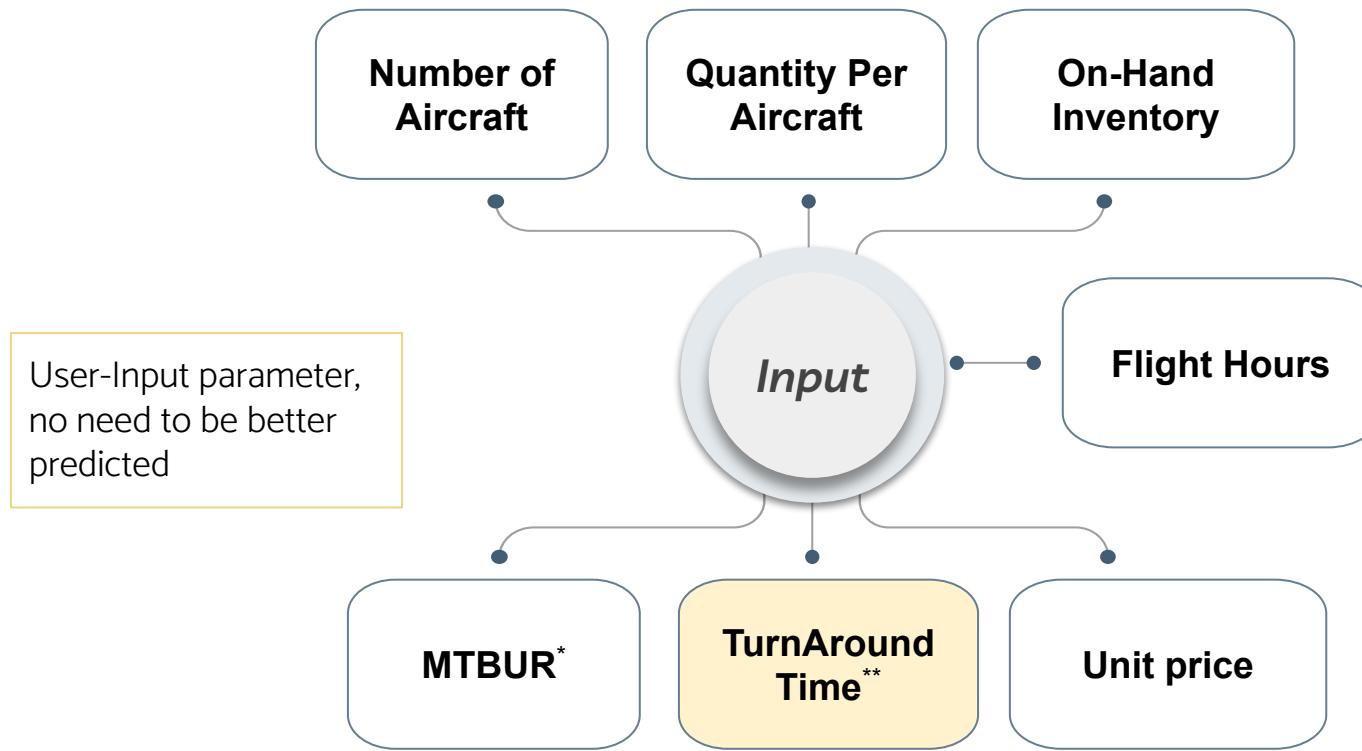
Improving Quality of Input Data



*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

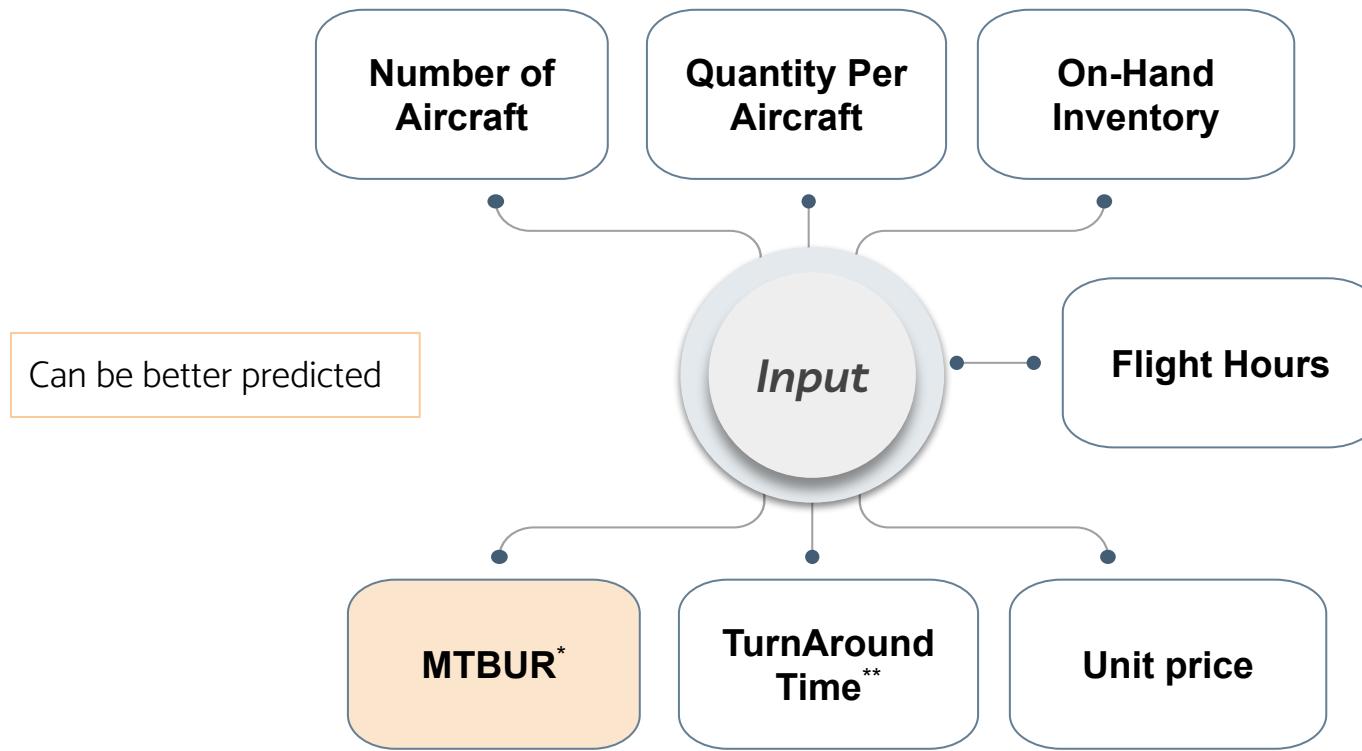
Improving Quality of Input Data



*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

Improving Quality of Input Data

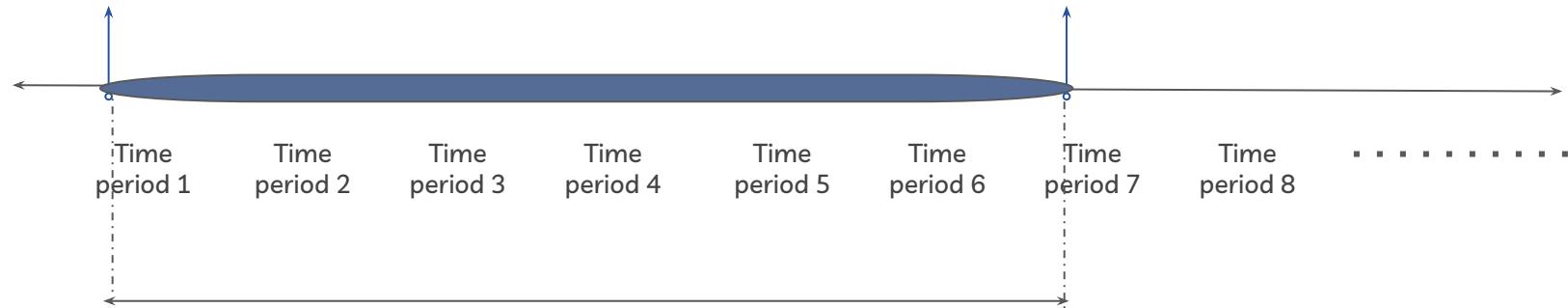


*MTBUR: Mean Time Before Unscheduled Removal;

**Turnaround Time includes the repair time and logistics/transportation time

Predicting MTBUR

Rotable i^{th} fitted onto an aircraft



**Mean Time Between
Unscheduled Removal
(MTBUR)**

Each rotatable type will have a MTBUR value of its own

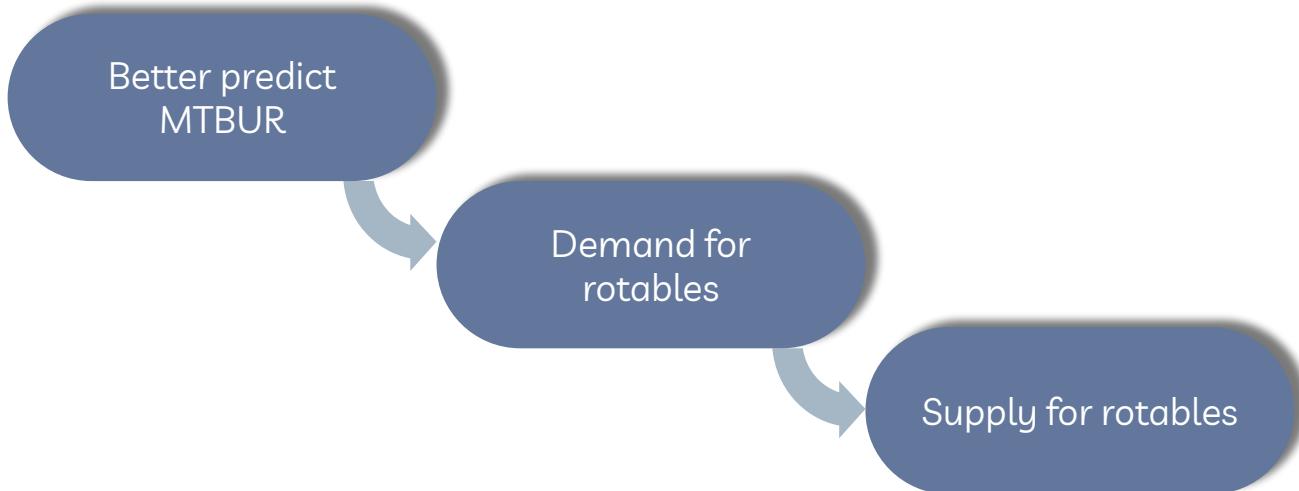
Predicting MTBUR



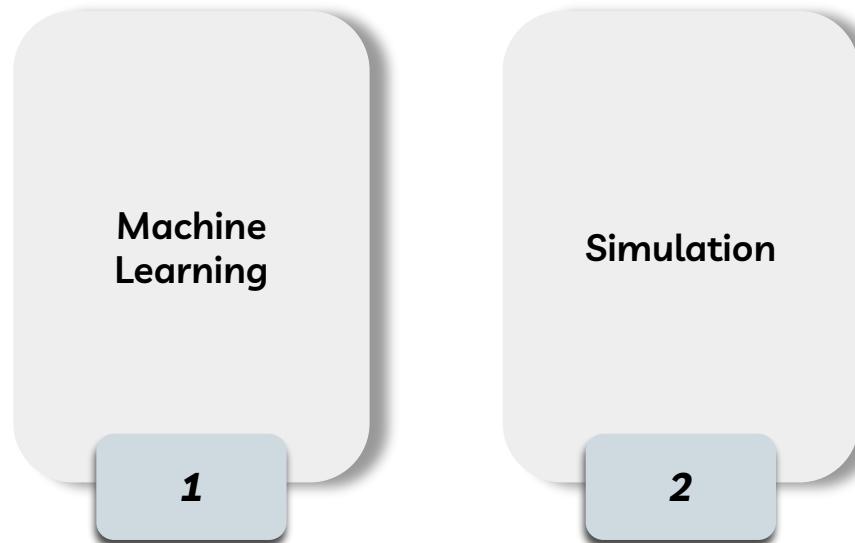
Majority of MTBUR used by SIAEC is provided by the original manufacturer which is **very generalized**¹ and **is skewed**, only a **minority** of rotatable parts have **historical MTBUR** available

¹ Generalized: For example, airplanes that frequently fly to more warm areas might corrode faster (hence lower MTBUR) as compared to cold climates, but these factors affecting the mTBUR are not captured in the aggregated MTBUR that is provided by the original rotatable manufacturer

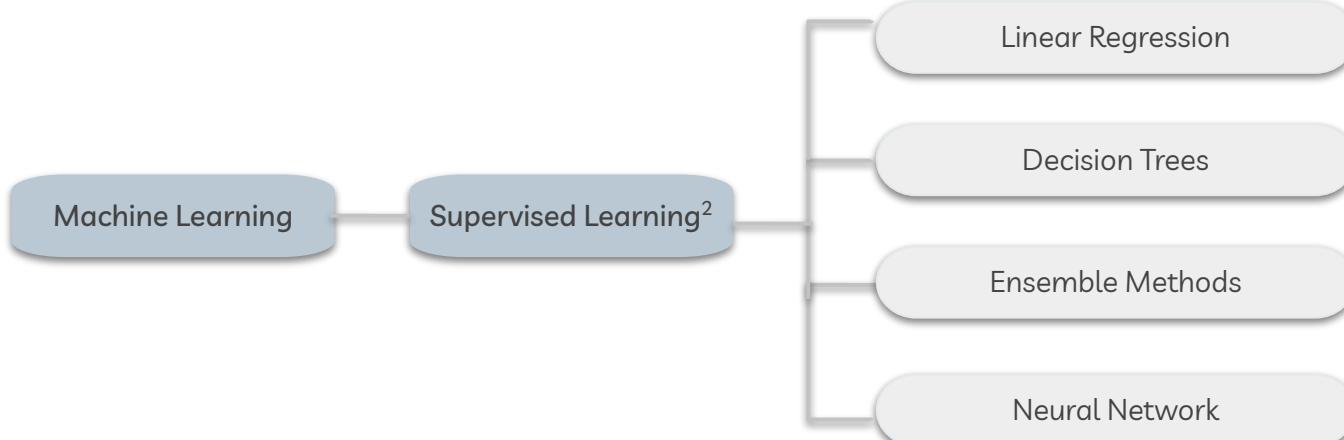
Predicting MTBUR



Predicting MTBUR



Machine Learning



Machine Learning



Factors affecting
MTBUR³

Running Life

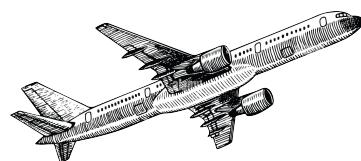
Machine usage

History of past failures

Number of past failures

Operating Conditions

Machine Learning



Factors affecting
MTBUR

Running Life

Machine usage

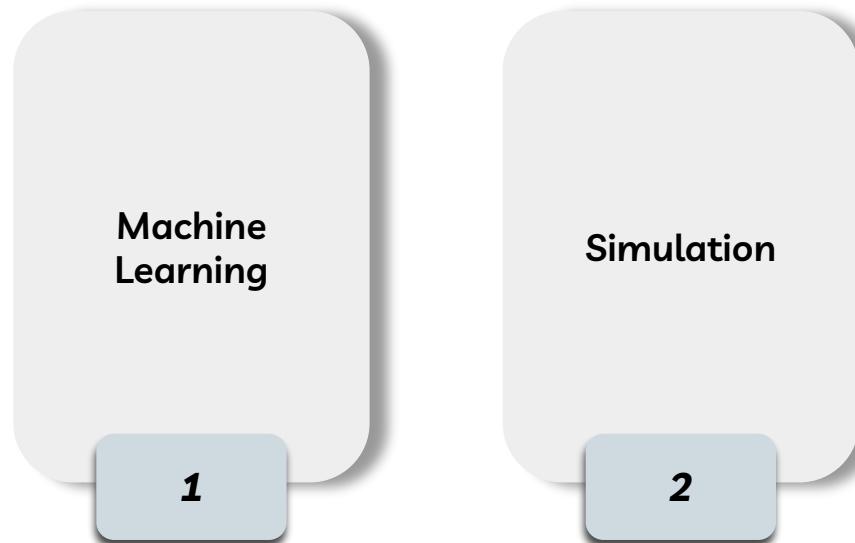
History of past failures

Number of past failures

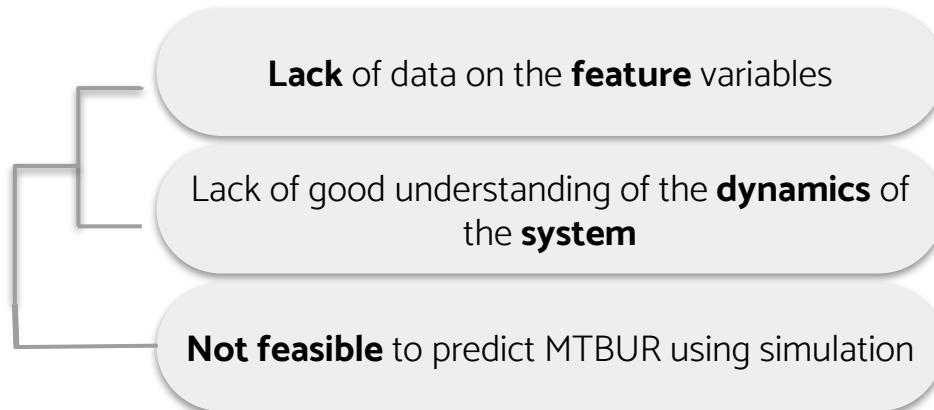
Operating Conditions

Lack of these data

Predicting MTBUR



Simulation⁴





Approach A

Optimization model

Long-term Provisioning

Approach B

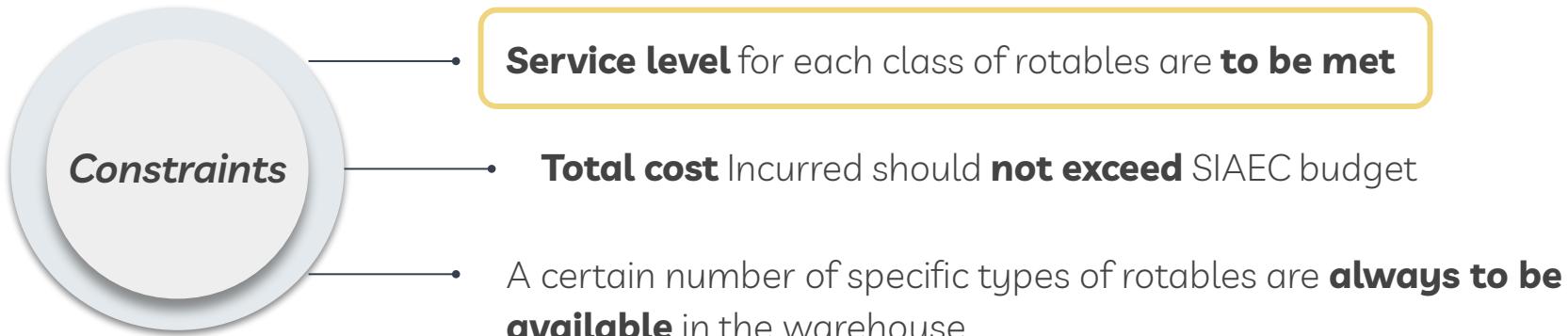
Improving Accuracy & Quality of
Input Data



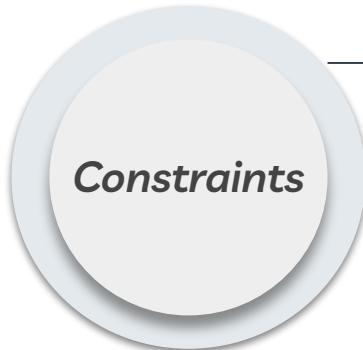
Approach C

Statistical Distribution of Demand

Statistical Distribution of Demand



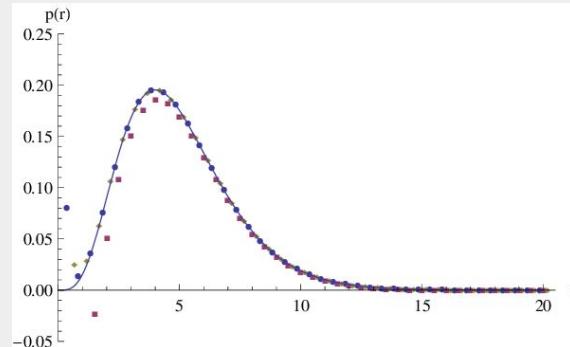
Statistical Distribution of Demand



Service level for each class of rotables are **to be met**



SIAEC assumes Poisson Distribution



Statistical Distribution of Demand

Alternative

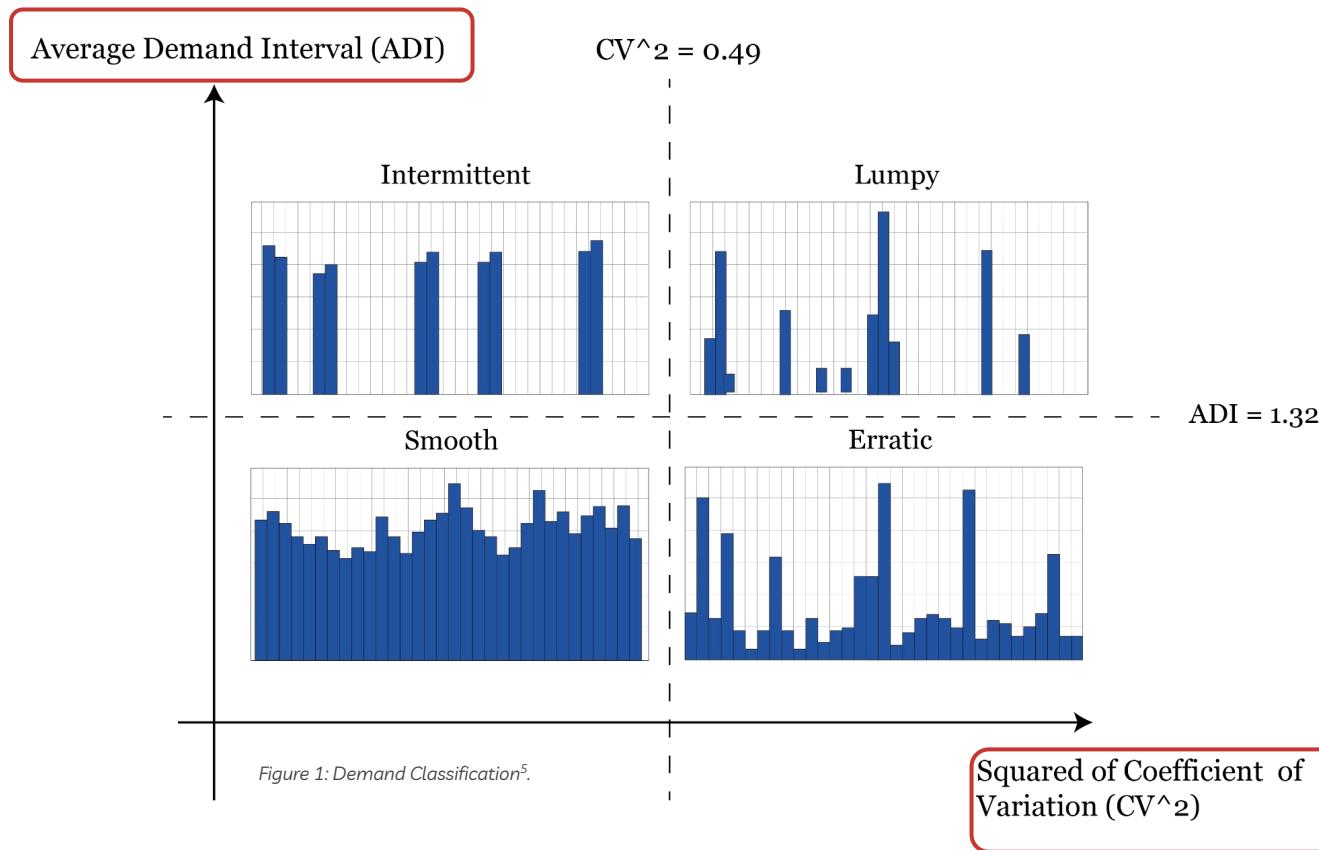
Compound

- Stuttering Poisson
- Negative Binomial

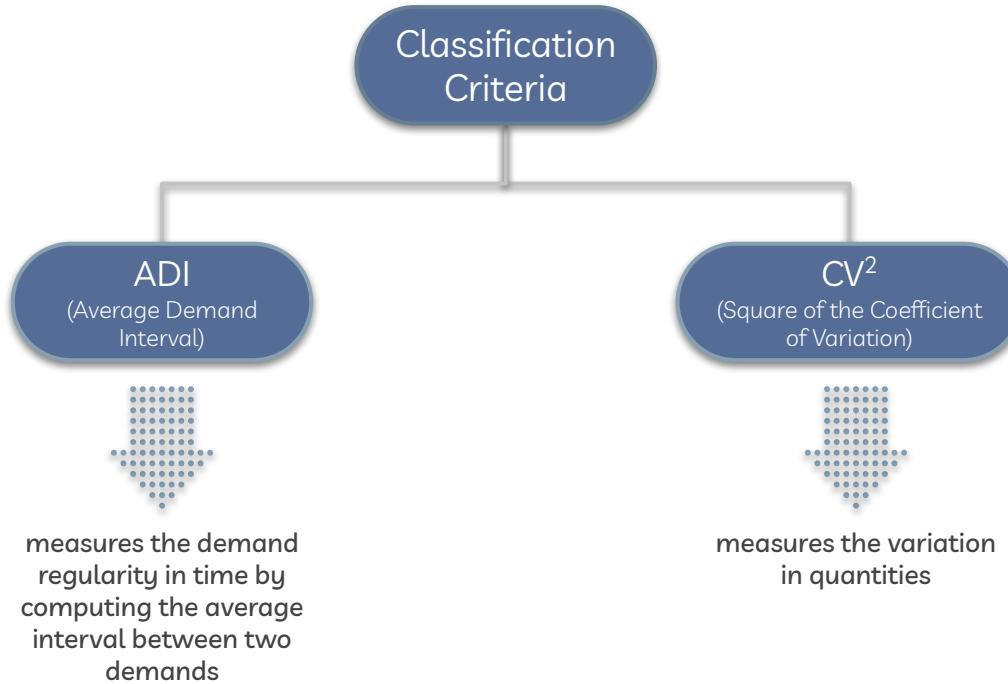
Non-Compound

- Normal
- Gamma

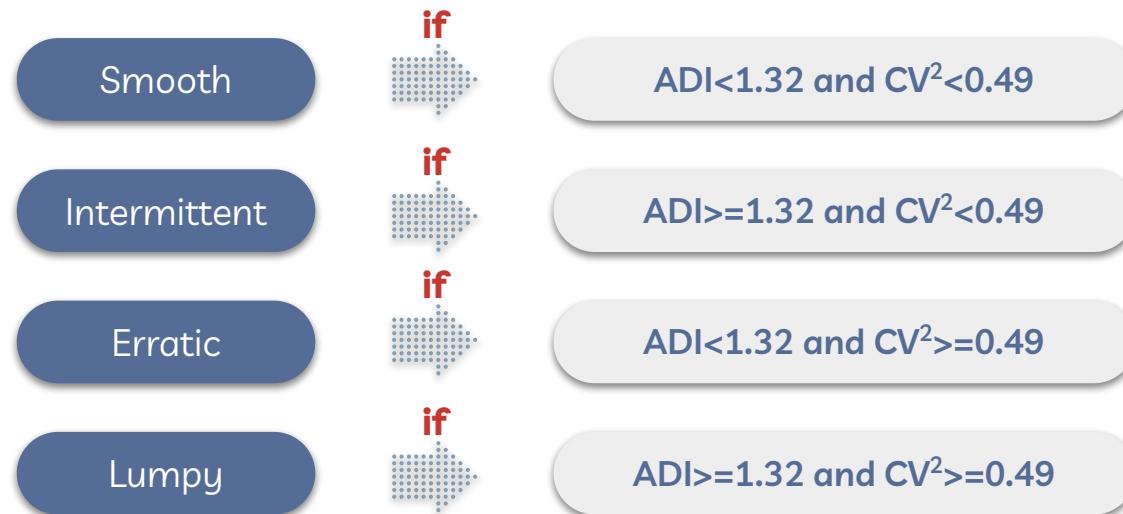
Statistical Distribution of Demand



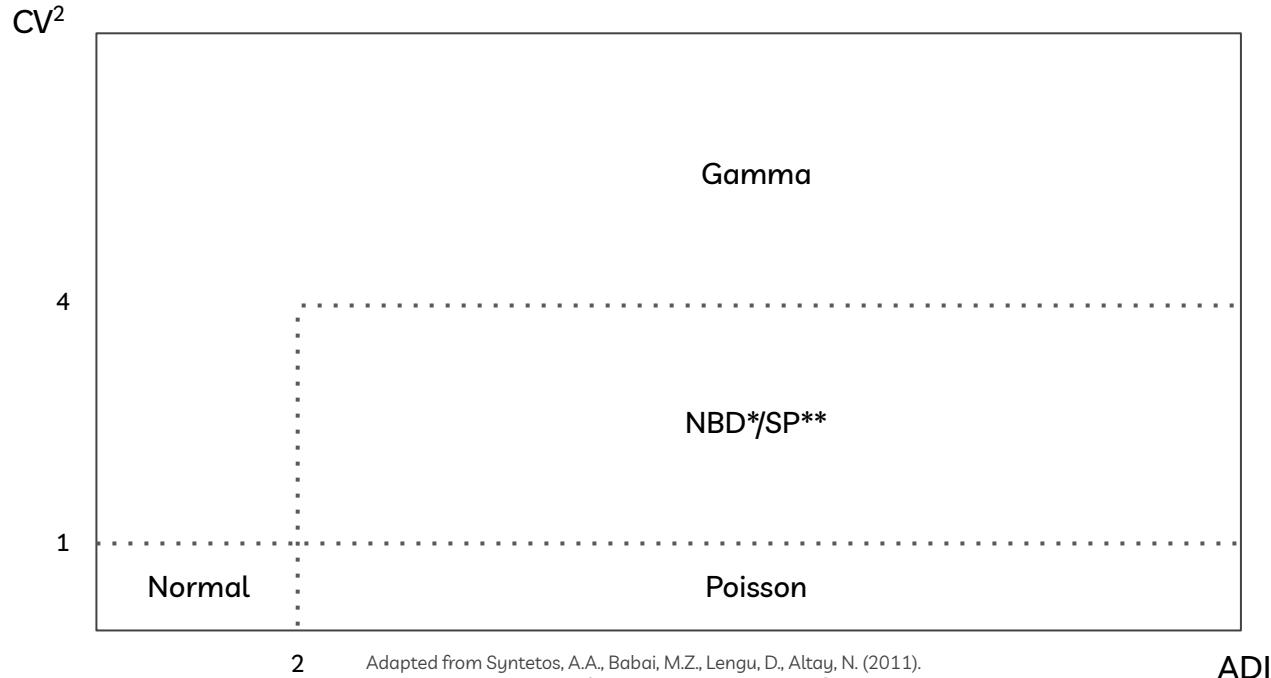
Statistical Distribution of Demand⁶



Statistical Distribution of Demand⁶



Statistical Distribution of Demand



* Negative Binomial Distribution

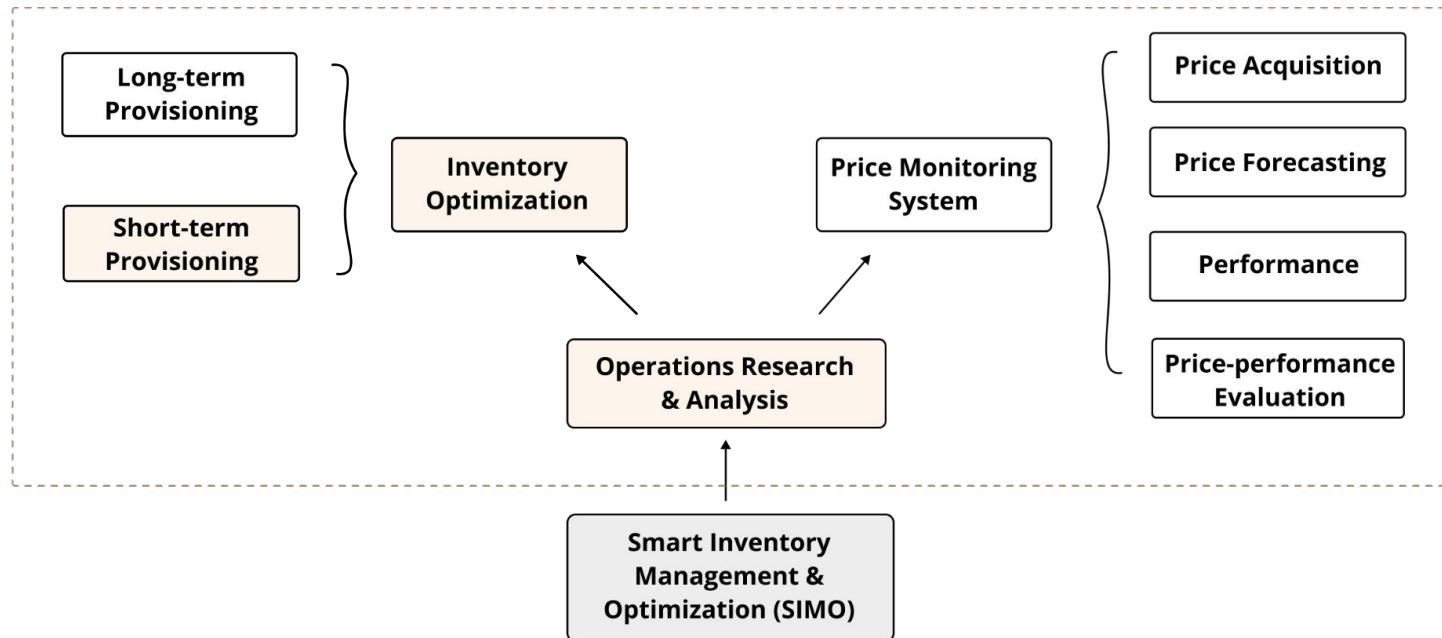
** Stuttering Poisson

2 Adapted from Syntetos, A.A., Babai, M.Z., Lengu, D., Altay, N. (2011). Distributional Assumptions for Parametric Forecasting of Intermittent Demand. In: Altay, N., Litteral, L. (eds) Service Parts Management. Springer, London. Distributional Assumptions for Parametric Forecasting of Intermittent Demand | SpringerLink





Short-term Provisioning

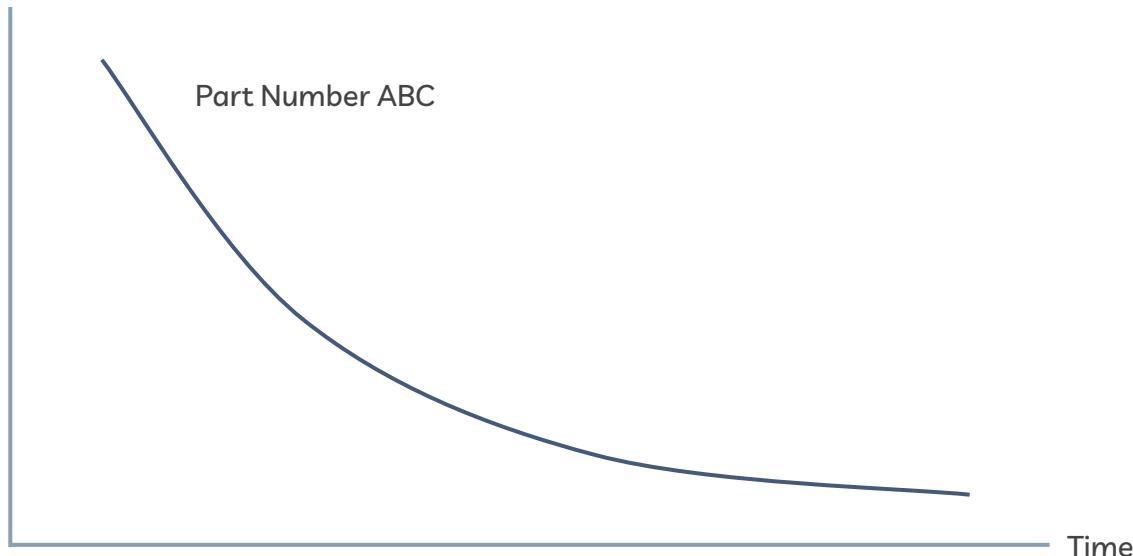


Milestone

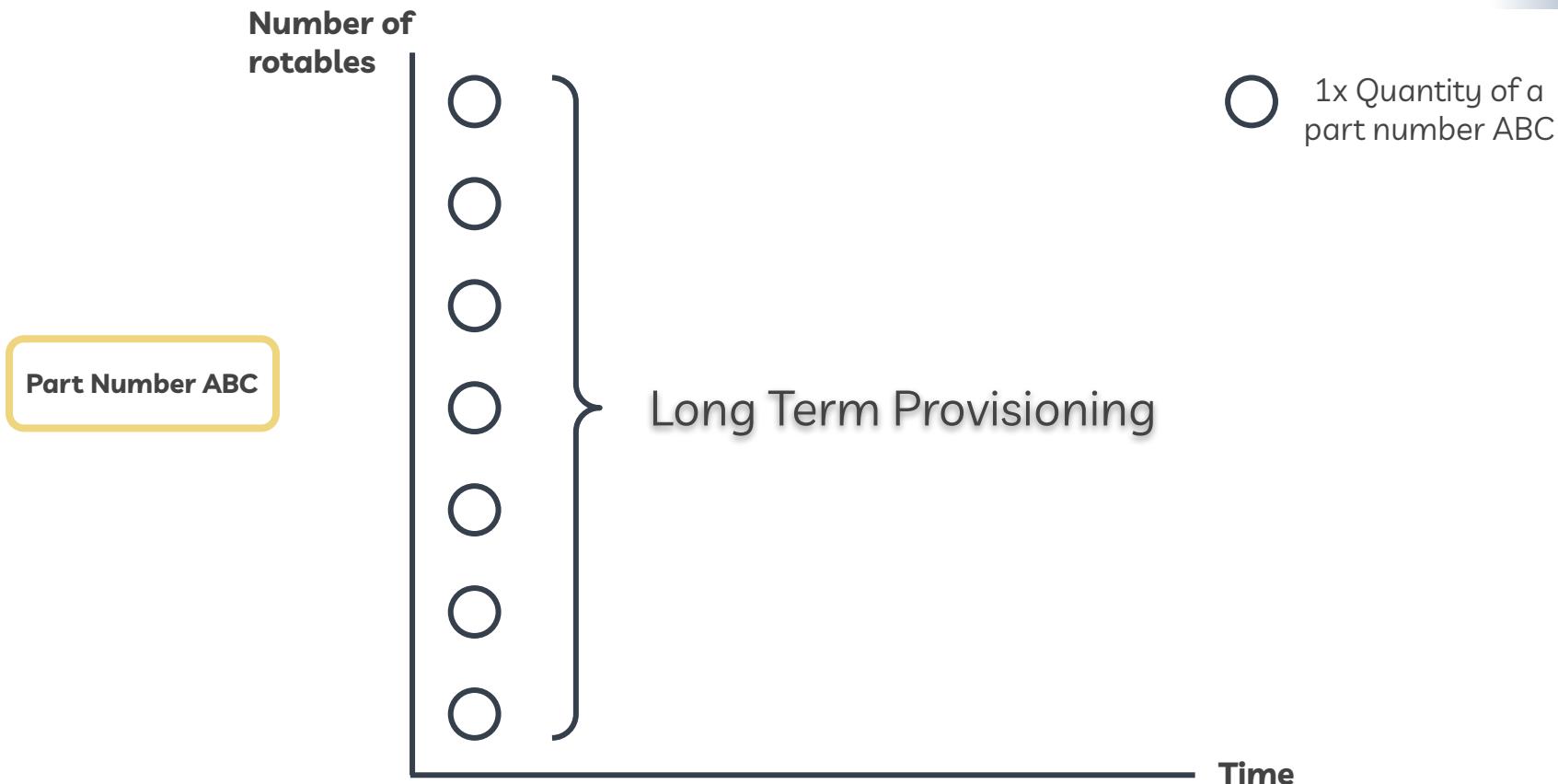


Short-term Provisioning

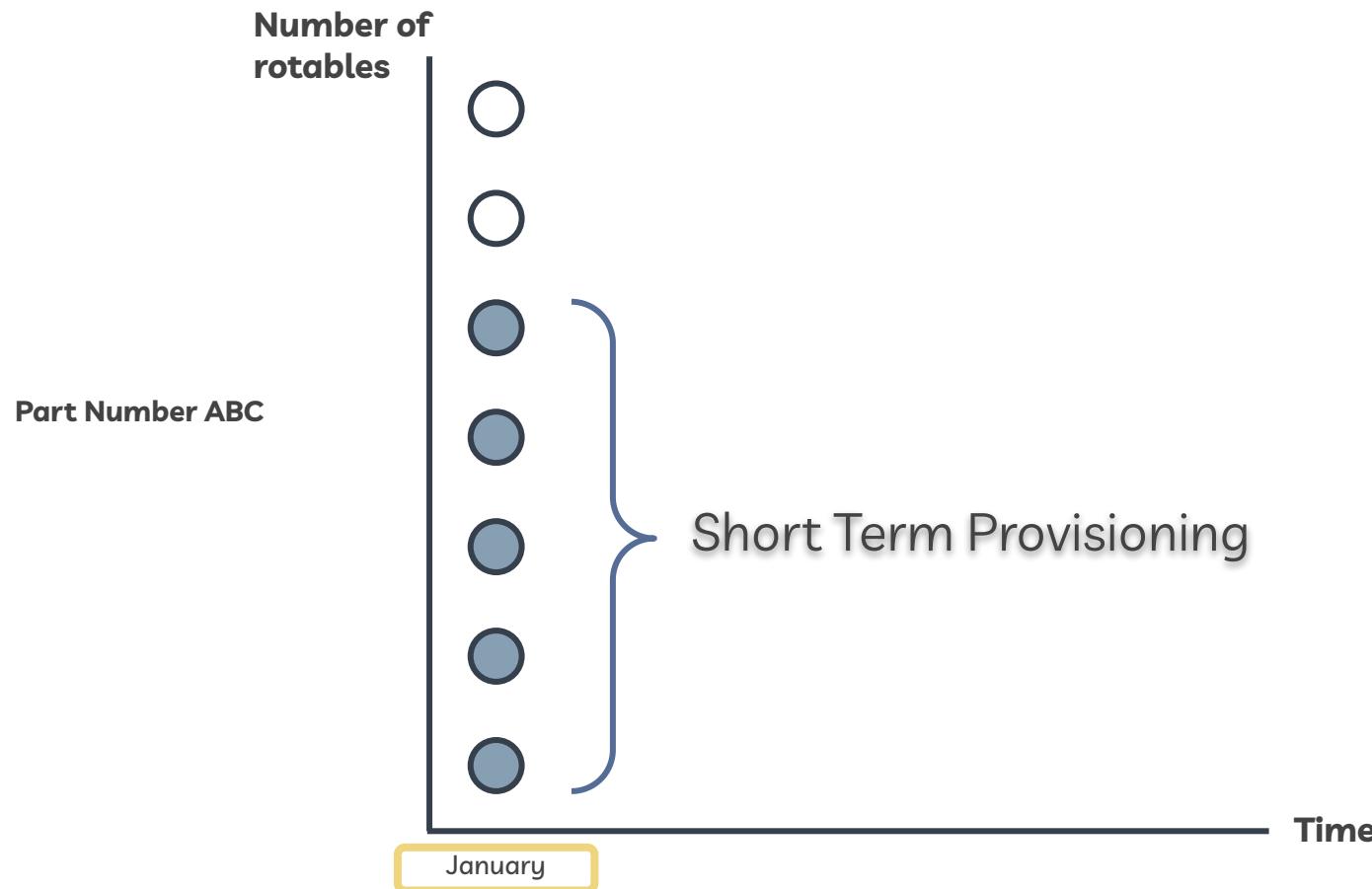
Rotables Condition



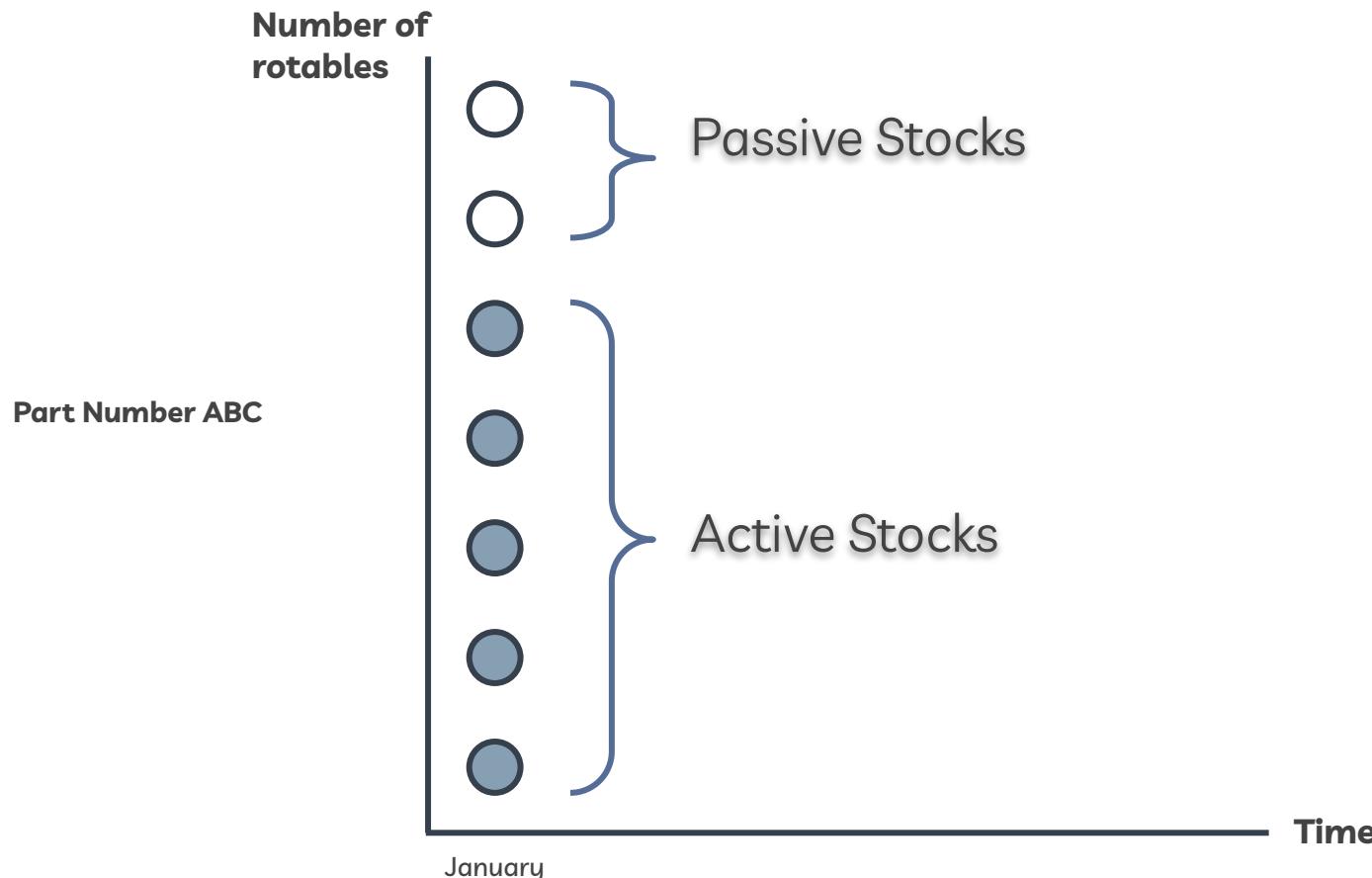
Short-term Provisioning



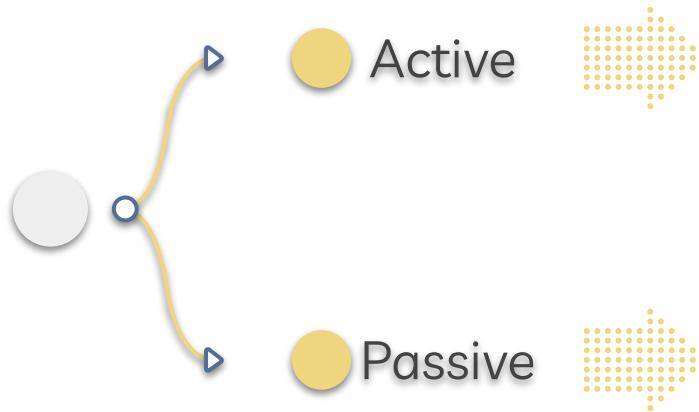
Short-term Provisioning



Short-term Provisioning



Short-term Provisioning



- Freshly Repaired
- Optimal Condition (Tip-top condition)

- To preserve rotables
- Slow down the deterioration rate

Short-term Provisioning

So what are our approaches to forecasting for active rotables?

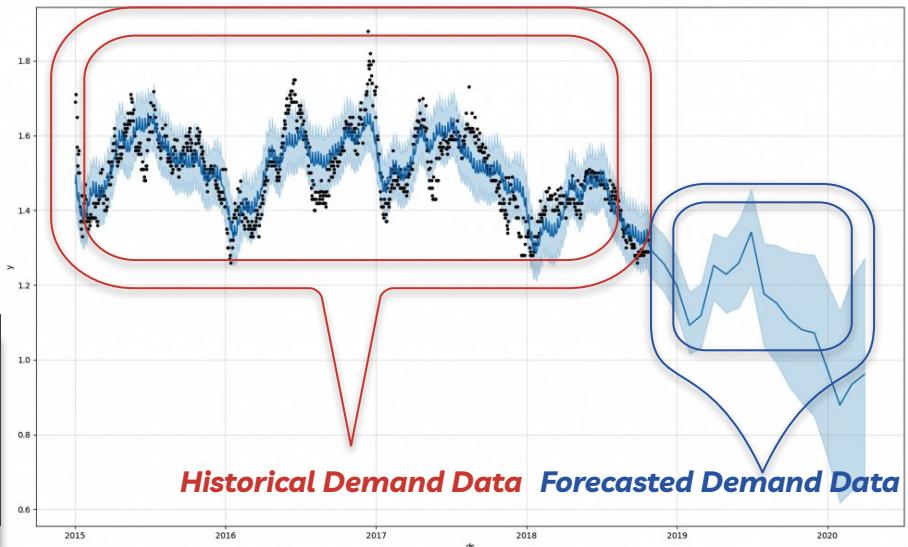
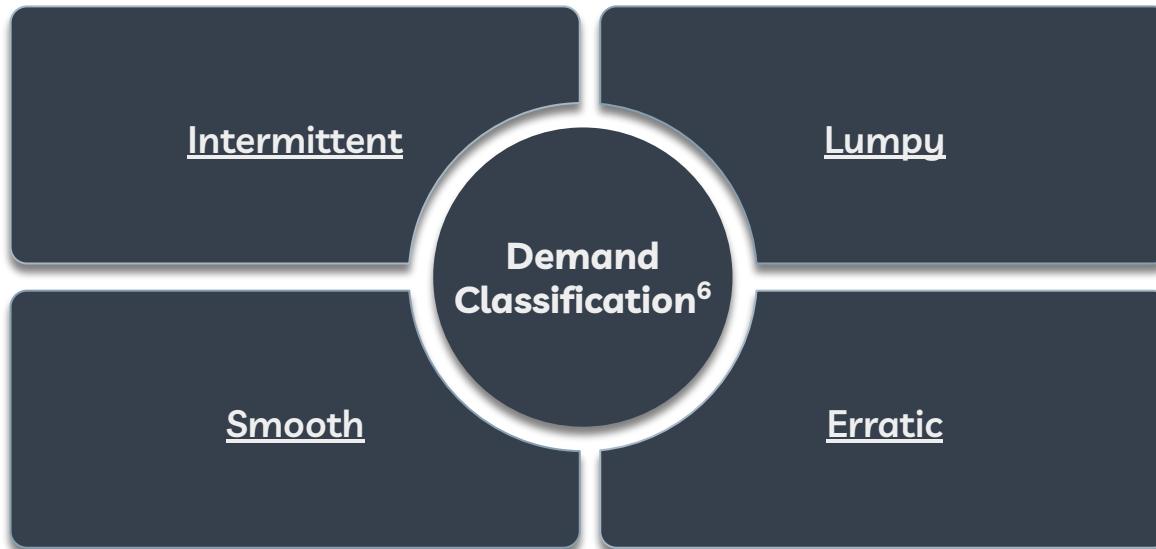
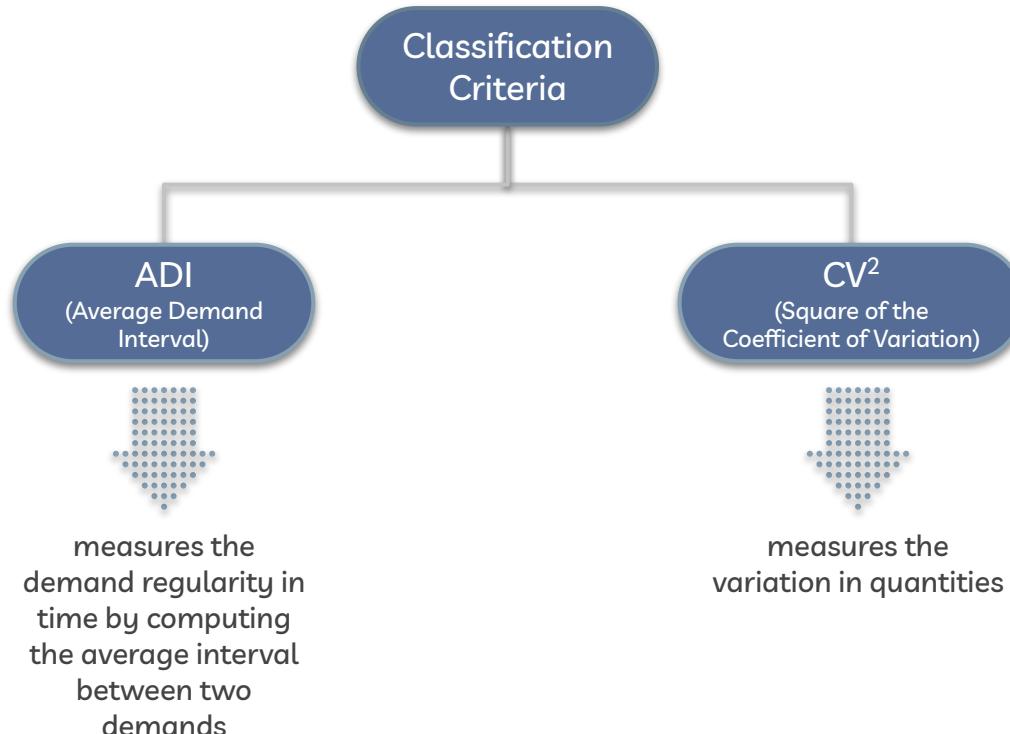


Figure 2: Time Series Forecasting. Adapted from Colin.catlin. (n.d.). Syllepsis. Retrieved April 12, 2022, from <https://syllepsis.live/2019/10/08/time-series-forecasting-a-quick-reference/>

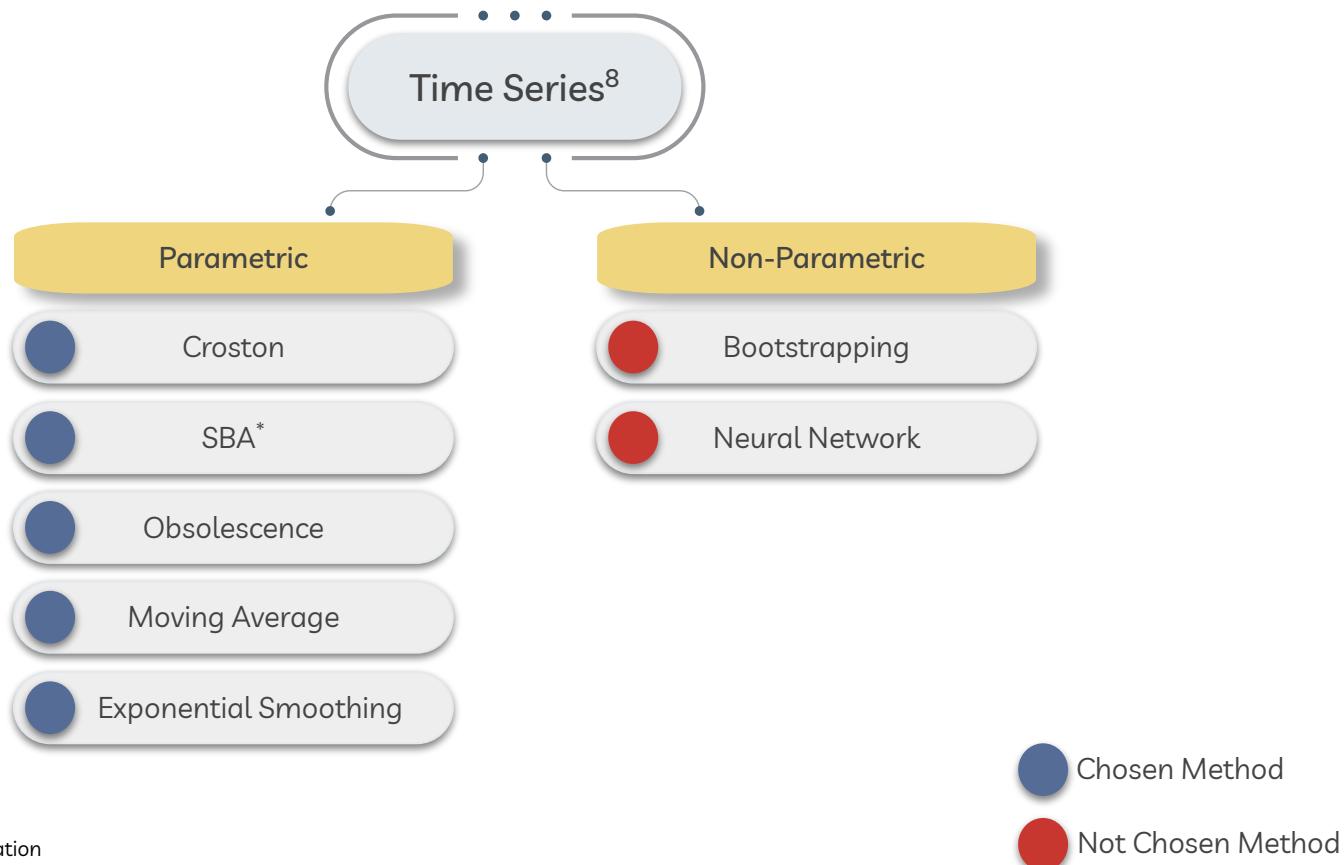
Short-term Provisioning



Short-term Provisioning



Short-term Provisioning



*. SBA: Syntetos–Boylan approximation

Short-term Provisioning

Design Criteria	Alternative Design Concepts	
	Parametric	Non-Parametric
Data Requirement	+	0
Accuracy	0	+
Time	+	0
Ease of implementation	+	0
Totals	+++	+

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.1

Price Monitoring System



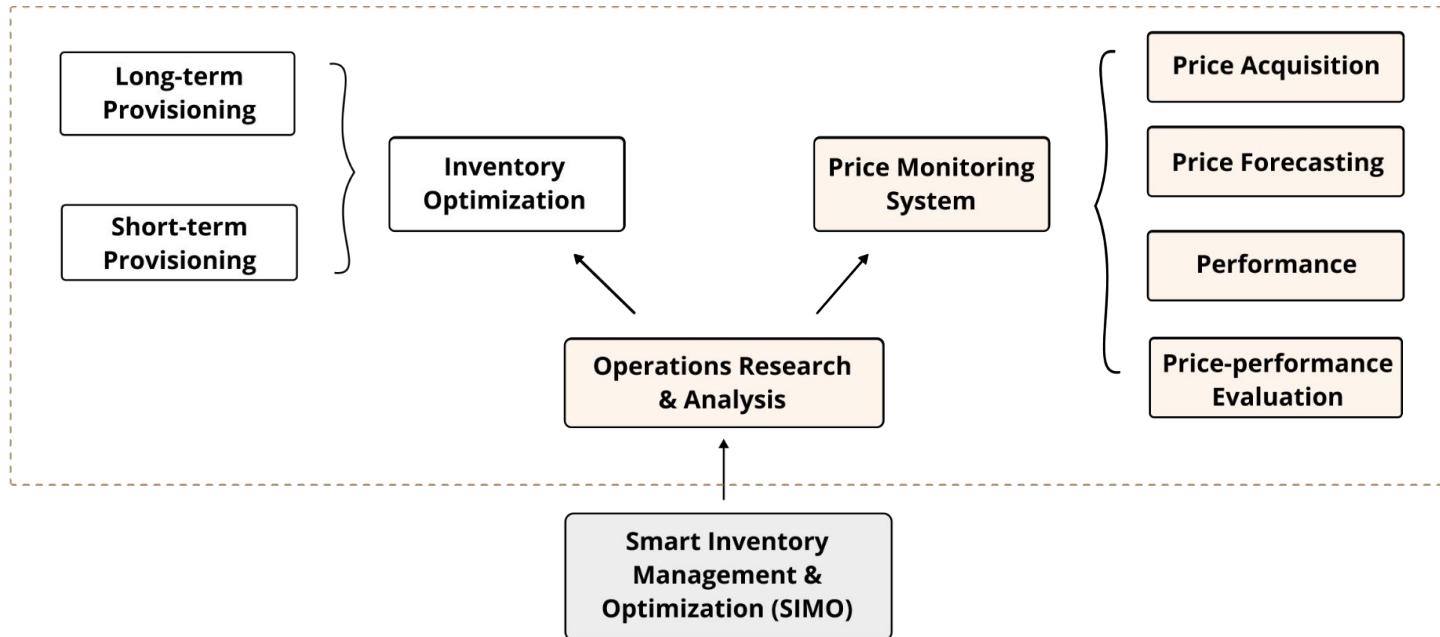
Short-term Provisioning



Short-term Provisioning



Price Monitoring System (Price Acquisition)



Price Monitoring System

- Price monitoring or price intelligence refers to the awareness of pricing in the market and the response to these changes in pricing.
- Metasearch engine⁹ → Sending queries through multiple search engines and aggregating these results
 - Ranking
 - Combine



Price Monitoring System



Price
Acquisition



Price
Forecasting



Performance
Evaluation

Price Monitoring System



Price
Acquisition

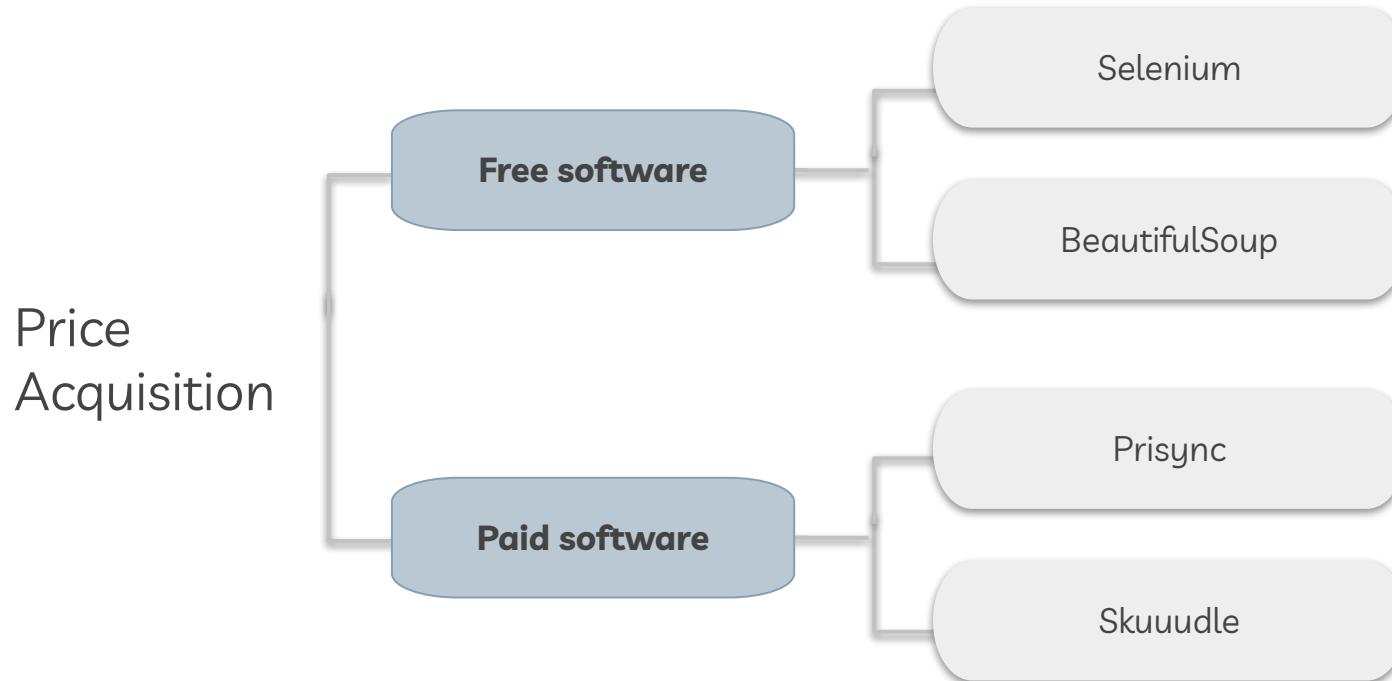


Price
Forecasting



Performance
Evaluation

Price Monitoring System



Price Monitoring System (Price Acquisition)

Design Criteria	Alternative Design Concepts			
	Skuuuudle	Prisync	BeautifulSoup	Selenium
Affordability	-	0	+	+
Ease of Integration	-	0	+	+
Adaptability	+	+	-	0
Total	-	+	+	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.2



Selenium

Price Monitoring System



Price
Acquisition

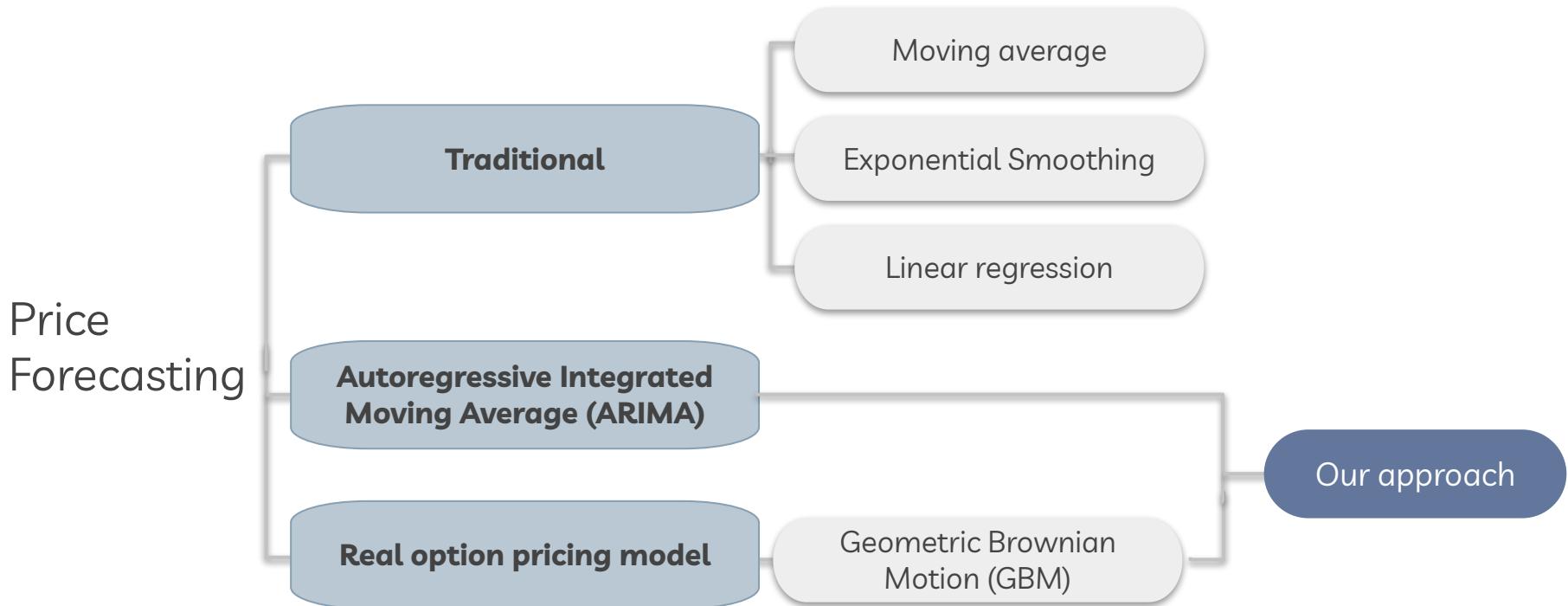


Price
Forecasting



Performance
Evaluation

Price Monitoring System (Price Forecasting)¹⁰



Price Monitoring System (Price Forecasting)¹⁰

Design Criteria	Alternative Design Concepts	
	Geometric Brownian Motion	ARIMA
Data Requirement	+	+
Accuracy	+	0
Time	0	0
Ease of implementation	+	+
Totals	+++	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.3

Price Monitoring System



Price
Acquisition

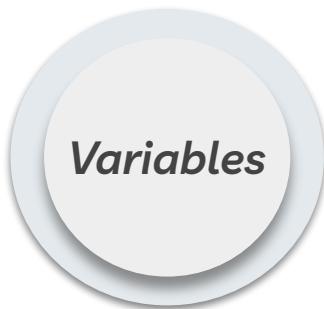


Price
Forecasting



Performance
Evaluation

Price Monitoring System (Performance)

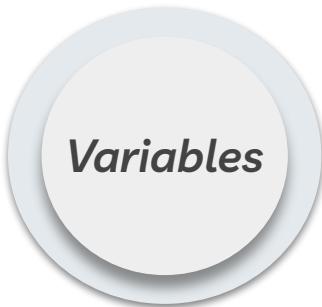


Time Lasted on Wing (TSI) across each vendor

Original Equipment Manufacturer (OEM) MTBUR

$$\frac{TSI}{MTBUR}$$

Price Monitoring System (Price to Performance)



MTBUR v.s Time Lasted on Wing (TSI)

Price given by OEMs v.s. Prices quoted

$$\frac{MTBUR}{\text{Price given by OEMs}}$$

Benchmark

$$\frac{TSI}{\text{Price quoted}}$$

Price Monitoring System



Price
Acquisition

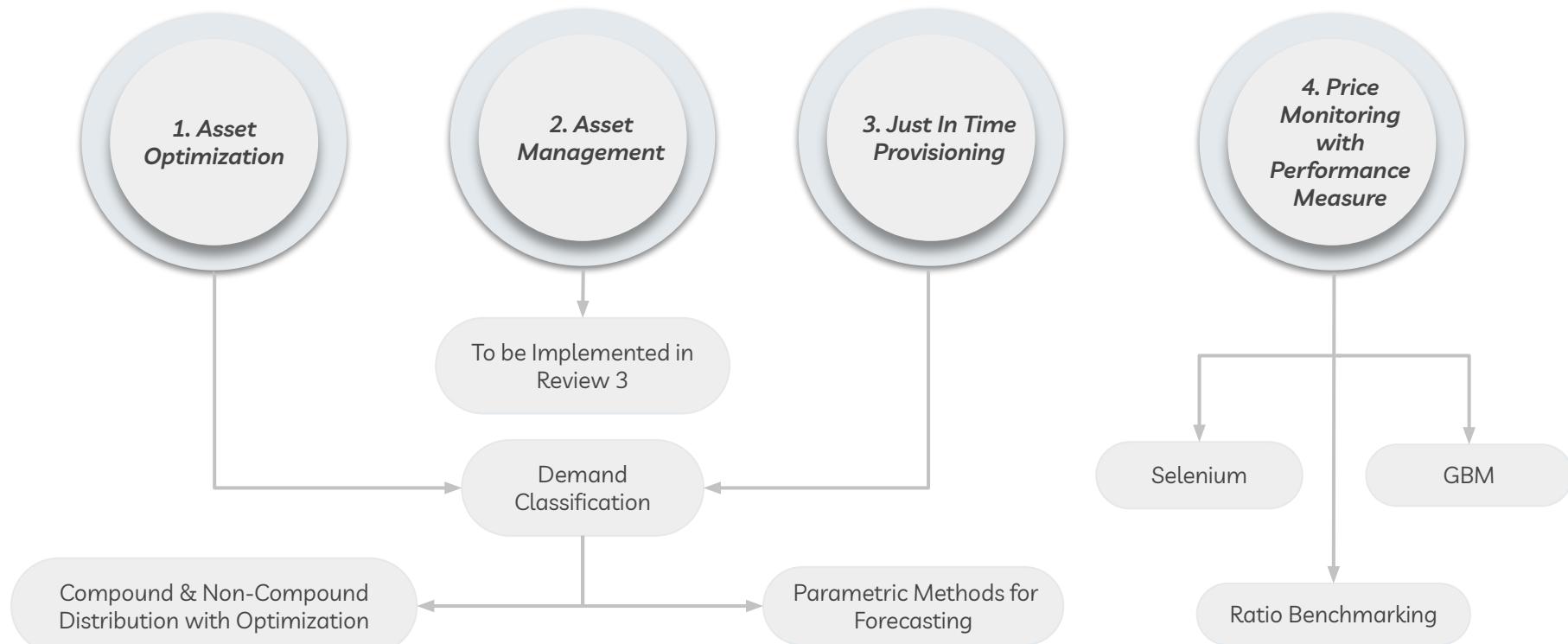


Price
Forecasting

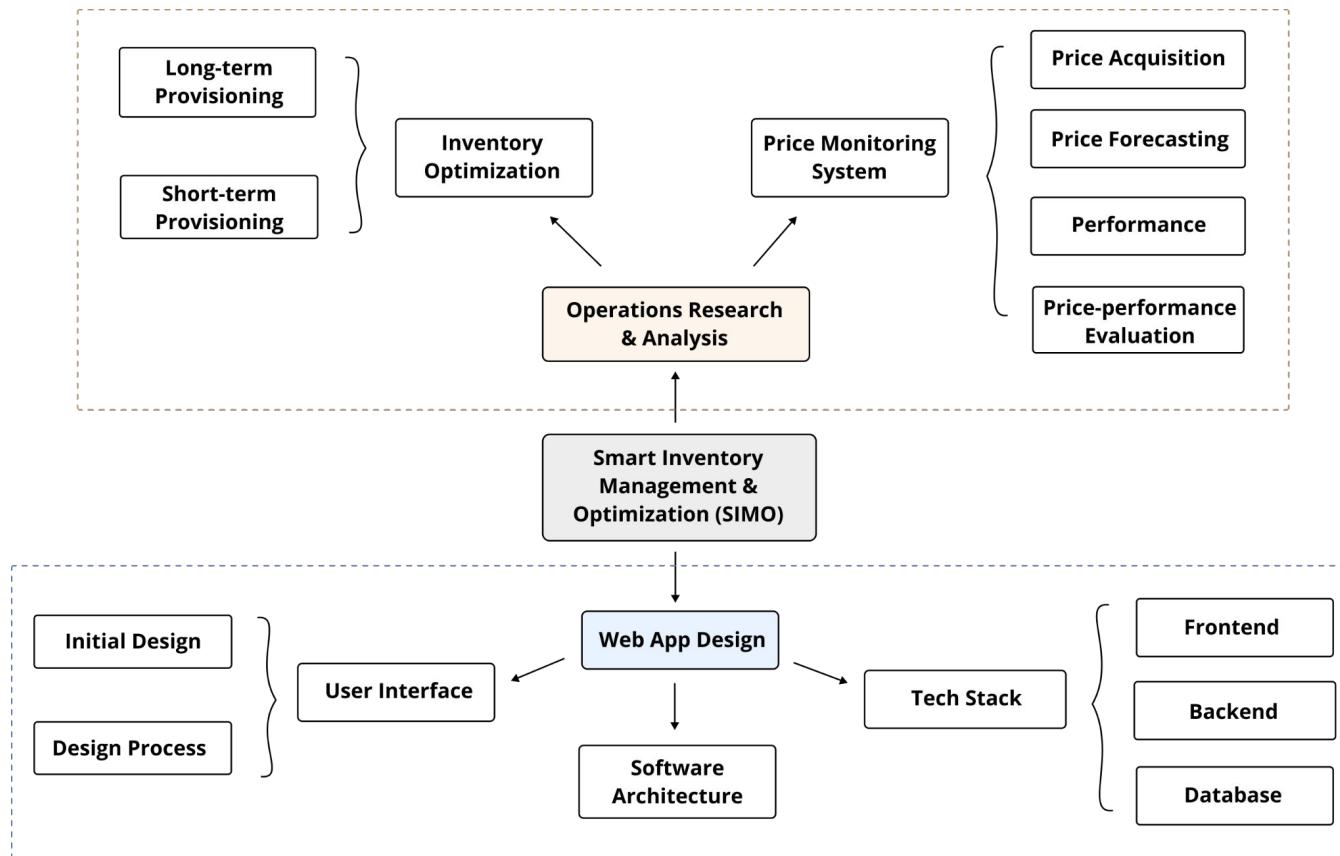


Performance
Evaluation

Summary



Concept Generation



Web Application Design

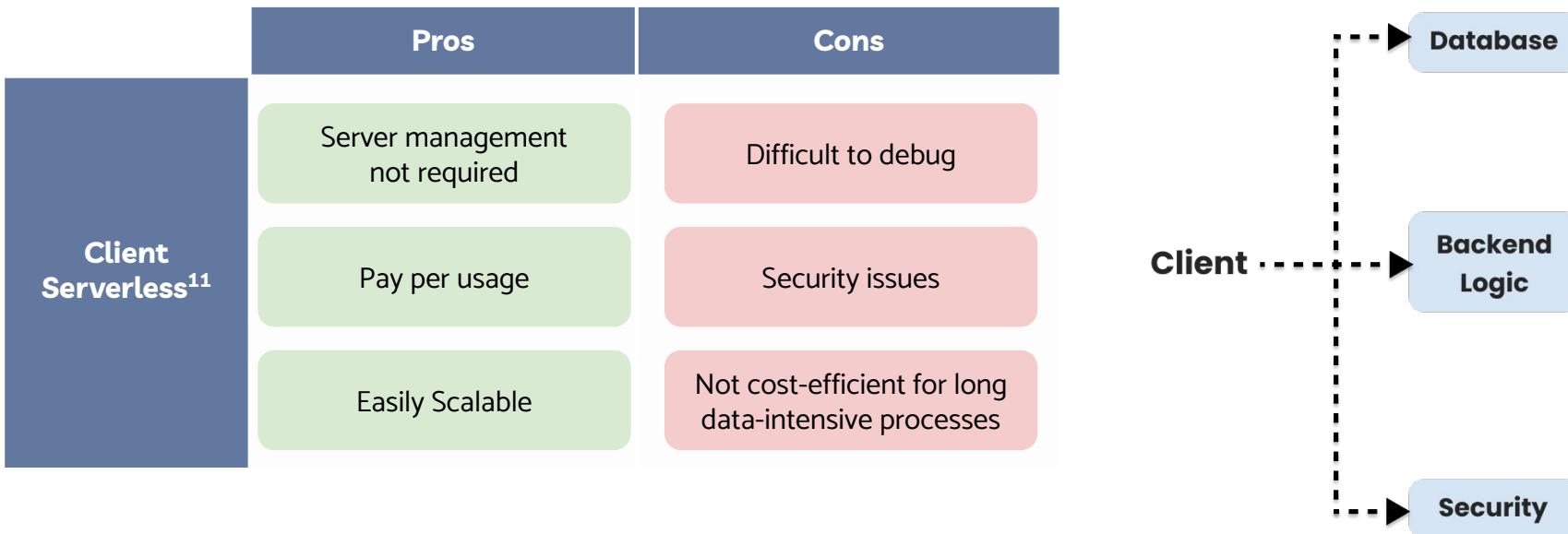
Software Architecture
User Interface
Tech stack

Software Architecture

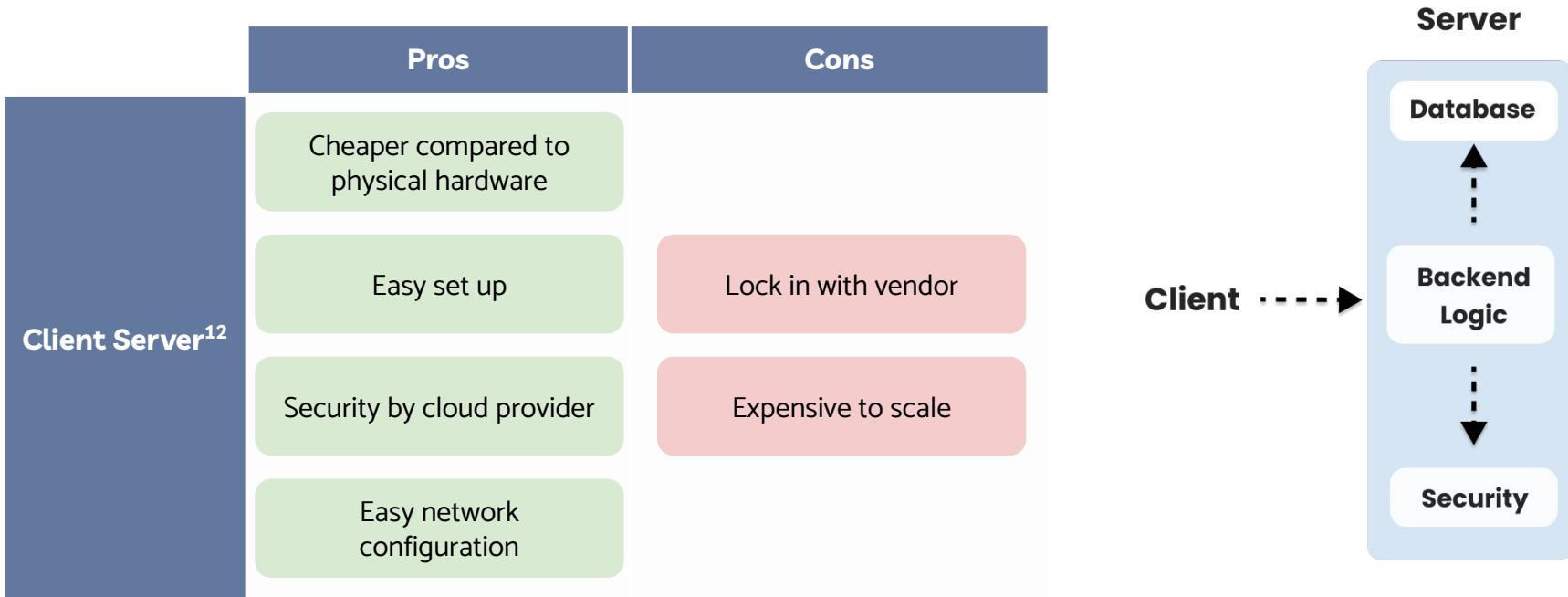
Client Serverless

Client Server

Software Architecture



Software Architecture

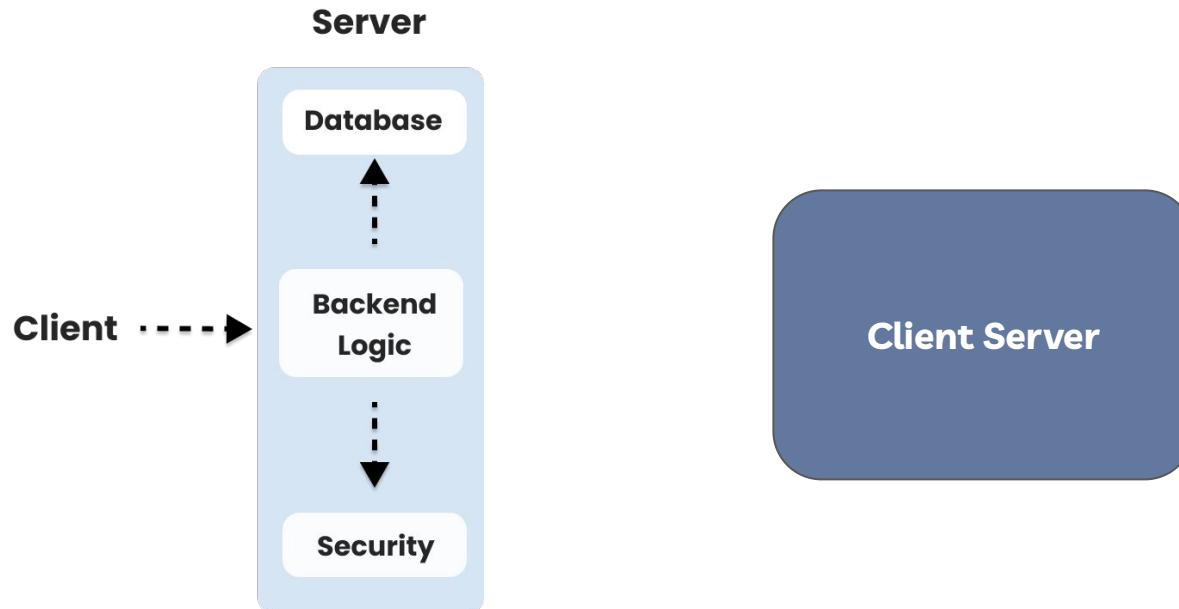


Software Architecture

Client Serverless

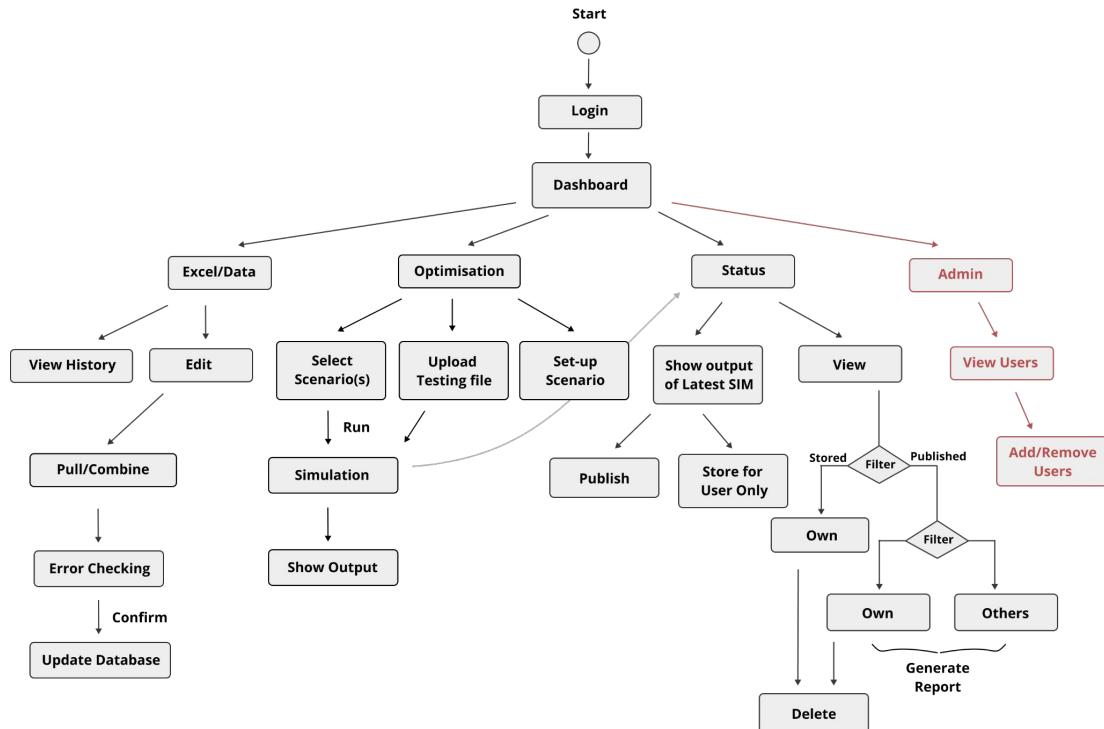
Client Server

Software Architecture



User Interface

User Interface



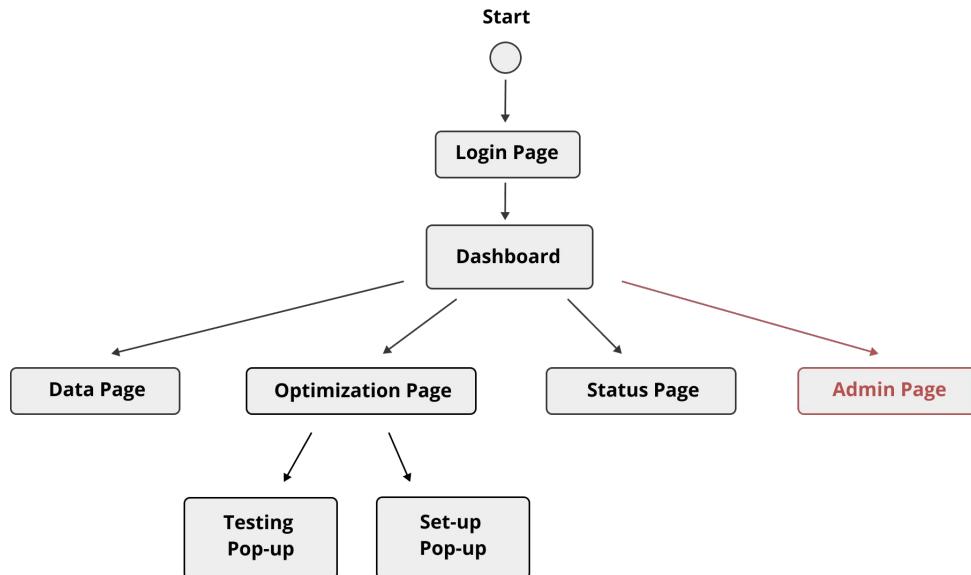
Action Flow Diagram

Contextualise user's needs

Understand user's workflow

Visualise functionalities

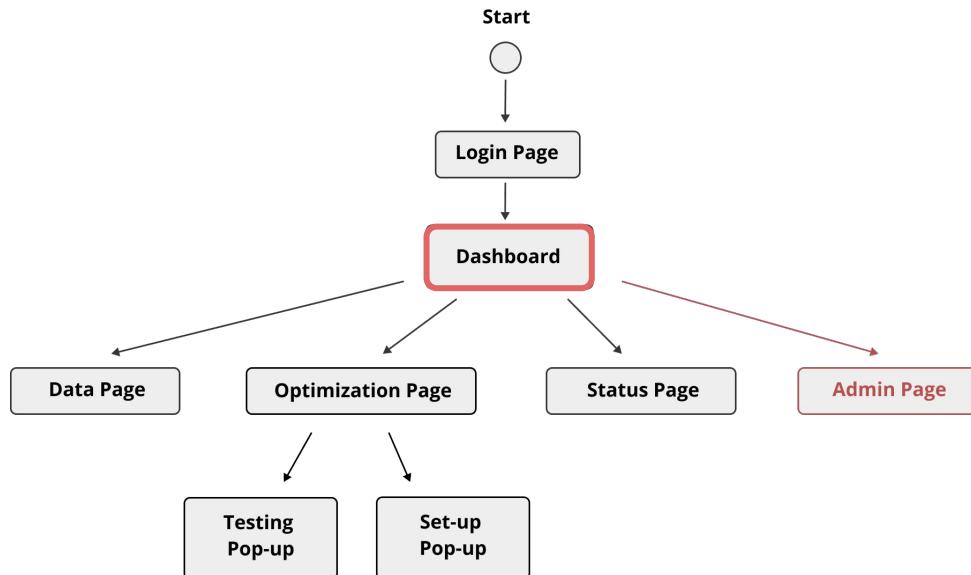
User Interface



Page Flow

Dashboard page
Data page
Optimization Page
Status Page
Admin Page (for admin user only)

User Interface



Dashboard

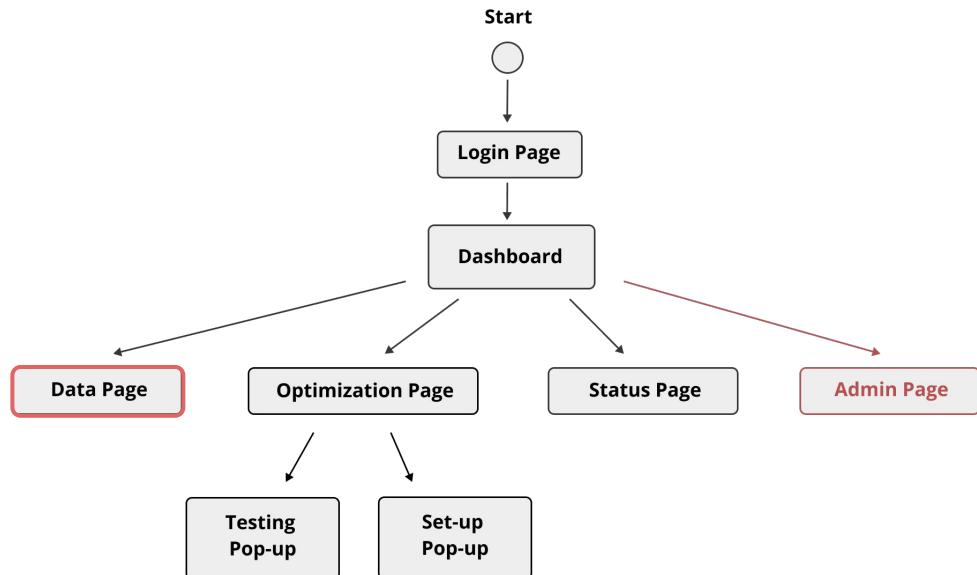
Inventory Management

Asset Management

Live Price Monitoring

Report and recommendation

User Interface

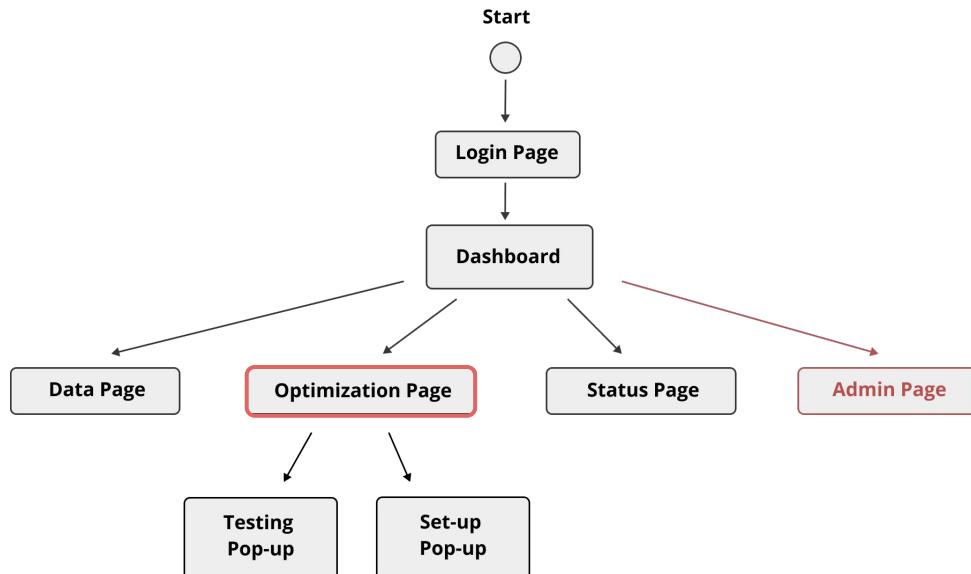


Data Page

Data Management

Update data file
View/Edit data file
Export data file

User Interface



Optimization Page

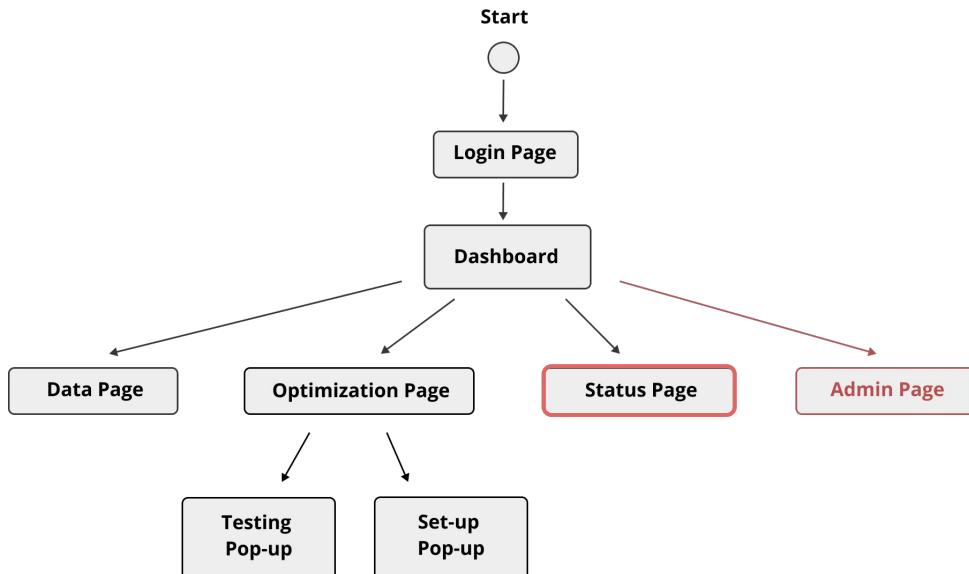
Simulation Model

Select Scenario
Set Constraints
Save Simulation output
Testing on temporary file

Supporting Features

Set new Scenarios
Adjust parameter values

User Interface



Status Page

Managing Reports

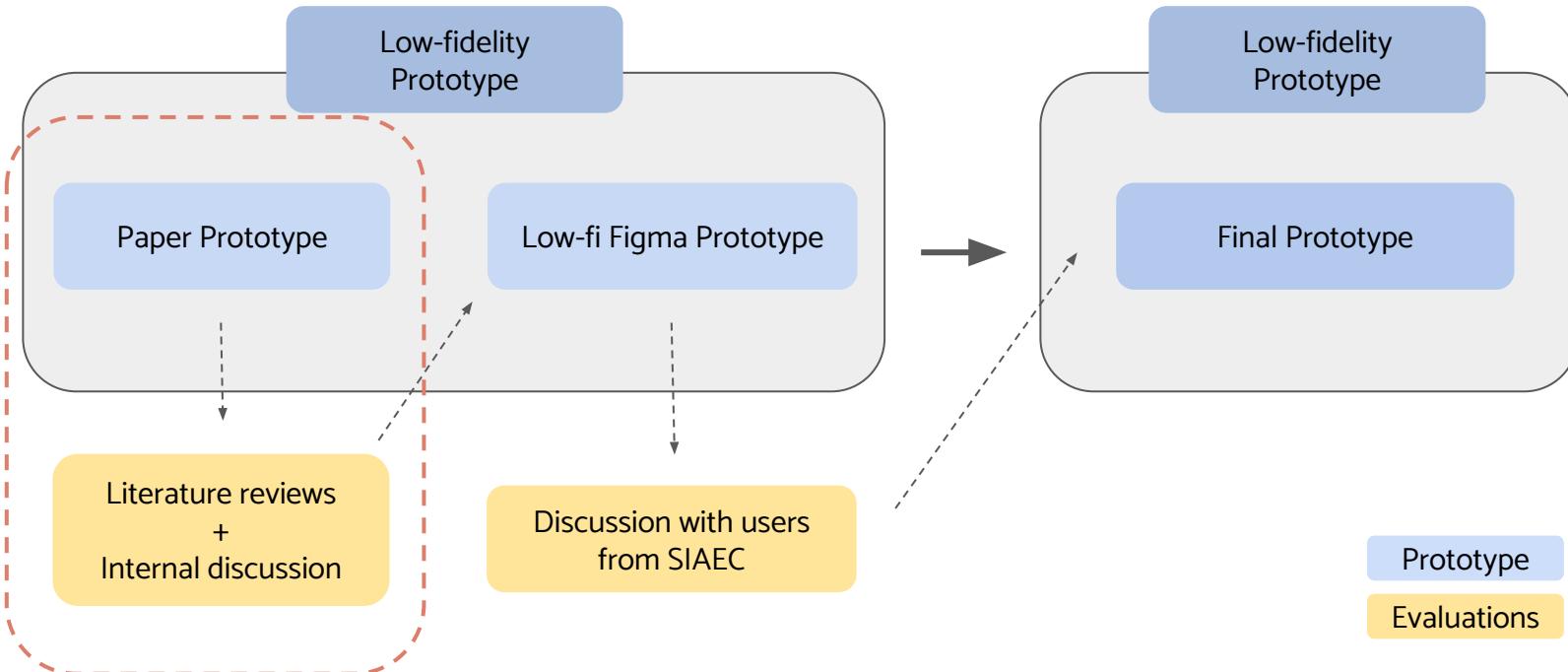
View/delete/publish report
Export report files

Sharing of Results

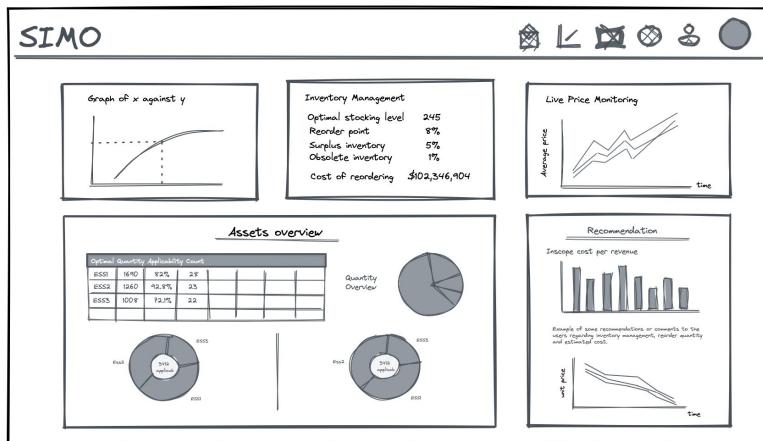
Export published report files
View published reports by other users

User Interface

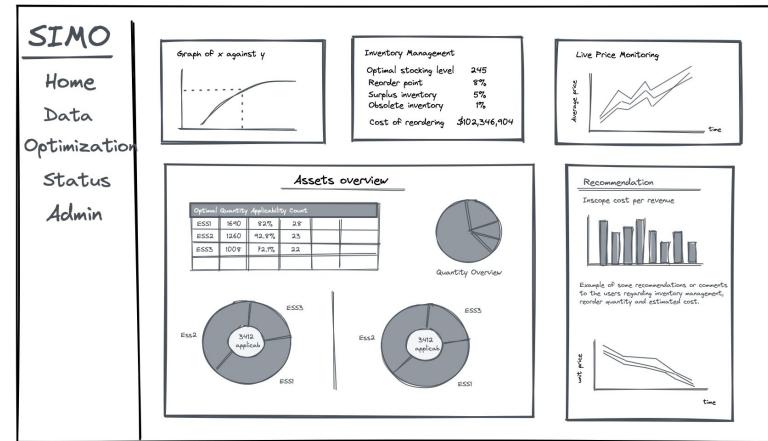
Prototypes Iterations



Navigation Bar¹³



Top Navigation Bar

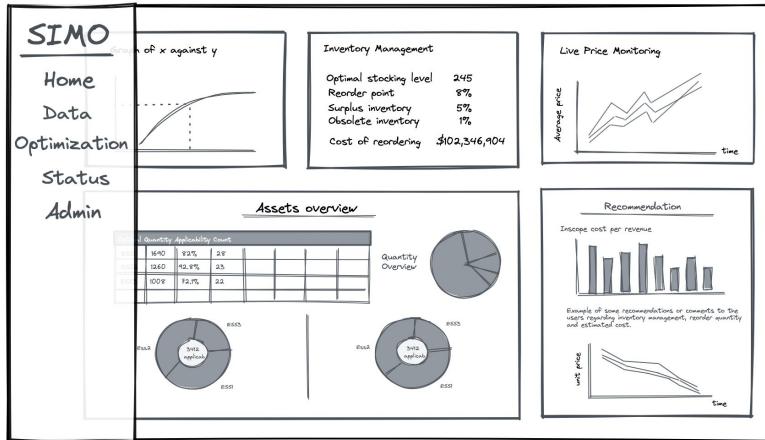


Left Navigation Bar

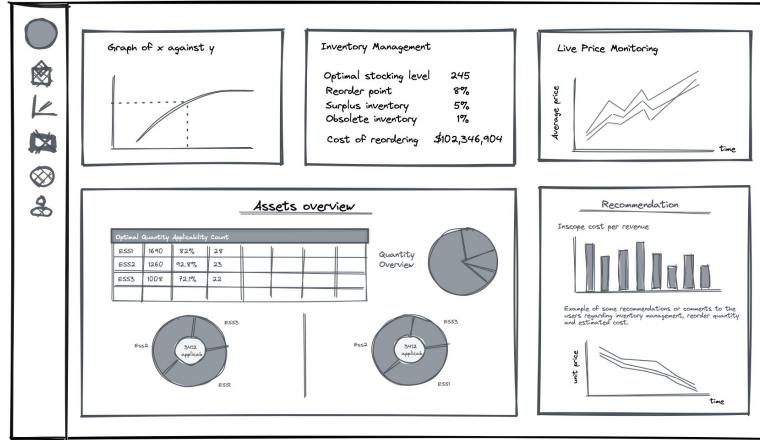
User Interface

Navigation Bar

Open



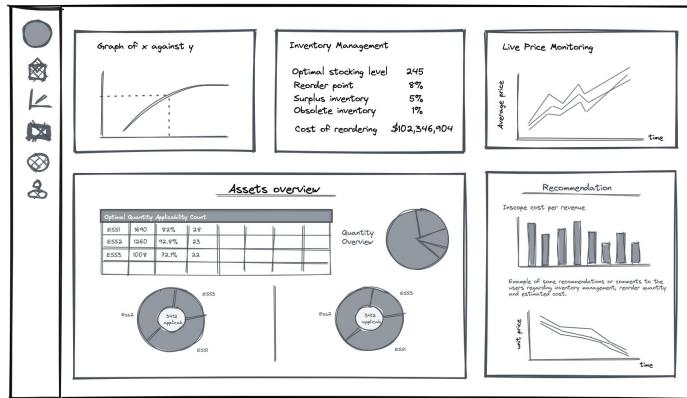
Collapsed



Collapsible Left Bar

User Interface

Improved Dashboard



Conceptual Model

Explore Various Alternative

Consolidate Screen Contents

Functionality

Navigation & Task flow

Terminology

Tech Stack

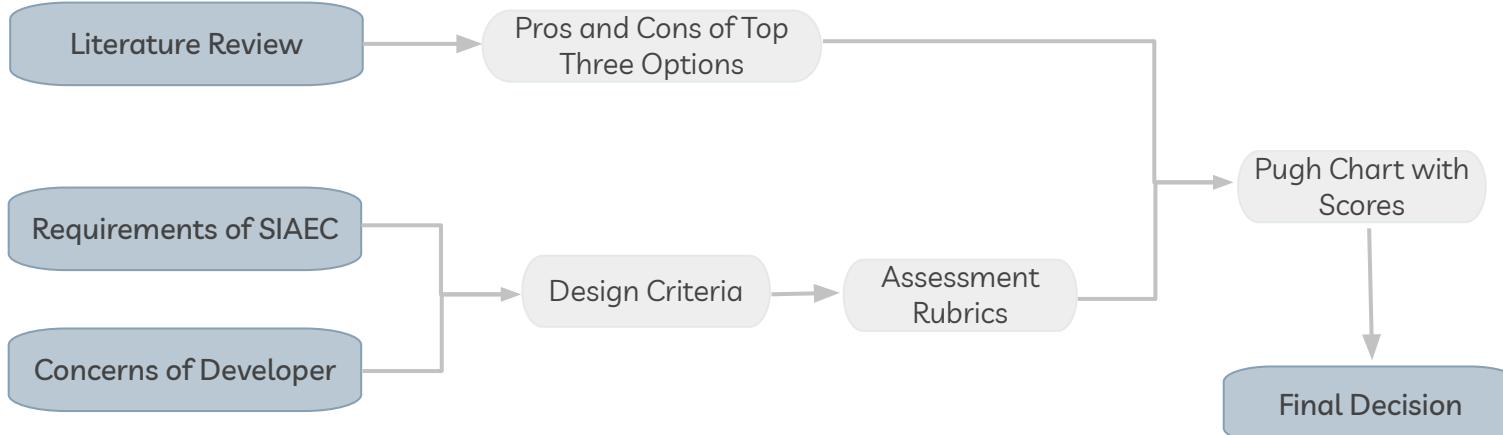
Frontend

Backend

Database Service

General Approach

For frontend, backend, and database exploration



Frontend Framework

Design Criteria	Alternative Design Concepts		
	React	Angular	Vue
Ease of Learning	+	-	-
Performance	+	-	+
Applicability of Features	+	+	0
Total	+++	-	0

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.4



React



ANGULAR



Vue.js

Frontend Framework

	React	Angular	Vue
Ease of Learning	+	-	-
Pros	Some team members have prior experience		
Cons	No team members has experience		

- Familiarity to framework

Some team members have prior experience

No team members has experience

No team members has experience

Pros

Cons

Frontend Framework

	React	Angular	Vue
Performance ¹⁴	+	-	+
Pros	Only re-render when needed Support server-side rendering Virtual DOM* increases performance	MVC structure* allows separation of tasks Slower response due to the number of features	Bundle size and Tree-shaking Code splitting
Cons			

Pros

Cons

* Virtual DOM: Virtual Document Object Model;

* MVC structure: Model-View-Controller Structure

Frontend Framework

	React	Angular	Vue
Applicability of Features ¹⁴	+	+	0
Pros	Reusable components	Modular development	Template Syntax
	Unidirectional data flow	Code conventions	Track dependencies
	Changeable state	Real-time synchronization	
Cons			

Pros

Cons

Frontend Framework

Design Criteria	Alternative Design Concepts		
	React	Angular	Vue
Ease of Learning	+	-	-
Performance	+	-	+
Applicability of Features	+	+	0
Total	+++	-	0

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.4



Backend Framework

Design Criteria	Alternative Design Concepts		
	Golang Gin	Express.js	Django
Ease of Learning	-	+	0
Ease of Implementation	0	-	+
Performance	0	-	+
Total	-	-	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.5



Express



django

Backend Framework

	Golang Gin	Express.js	Django
Ease of Learning	-	+	0

- Familiarity to framework
- Available learning resources

Relative small community

Available learning resources

Available learning resources

Relative bad documentation

Some team members have experience

No team members has experience

Pros

Cons

Backend Framework

	Golang Gin ¹⁵	Express.js ¹⁶	Django ¹⁷
Ease of Implementation	0	-	+
• Code size • Flexibility	Syntax is relatively easy to learn	Same language for both frontend and backend	Readymade packages
	Easy to create middlewares	Heavy code changes due to unstable API	Model View Template
	Low reusability of code due to no OOP*	Callback functions are hard to read but needed	Lack of code conventions

Pros

Cons

Backend Framework

	Golang Gin ¹⁸	Express.js ¹⁶	Django ¹⁹
Performance	0	-	+
• Response time	Radix tree	Event-driven	MTV Architecture
• Processing speed	Built to handle concurrent tasks	Unable to handle CPU intensive task	Same language as our optimization model
	Hard to integrate with optimization model (Python)		

Pros

Cons

Backend Framework

Design Criteria	Alternative Design Concepts		
	Golang Gin	Express.js	Django
Ease of Learning	-	+	0
Ease of Implementation	0	-	+
Performance	0	-	+
Total	-	-	++

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.5



Express



django

Database Service

Design Criteria	Alternative Design Concepts		
	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Performance	-	+	+
Applicability of Features	-	+	0
Cost-Efficiency	-	0	0
Total	---	++	+

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.6



Database Service

	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Performance ²⁰	-	+	+
Pros	Inefficient for complex queries	Instant provisioning and auto scaling with EC2*	Sharding for cloud-base storage
• Query speed • Concurrency capabilities			

Pros

Cons

Database Service

	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Applicability of Features ²¹	-	+	0
Pros	Non-Relational	Data Replication (High Availability)	Data Replication (High Availability)
Cons		Robust Security Support	Dynamic Schemas
		Convenient integration with AWS Redshift	Limited document size
		Only vertical scaling which is costly	Non-Relational

Database Service

	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Cost-Efficiency	-	0	0

- Monthly expenditure

> SGD 300

Between SGD 100 -300

Between SGD 300 - 100

Pros

Cons

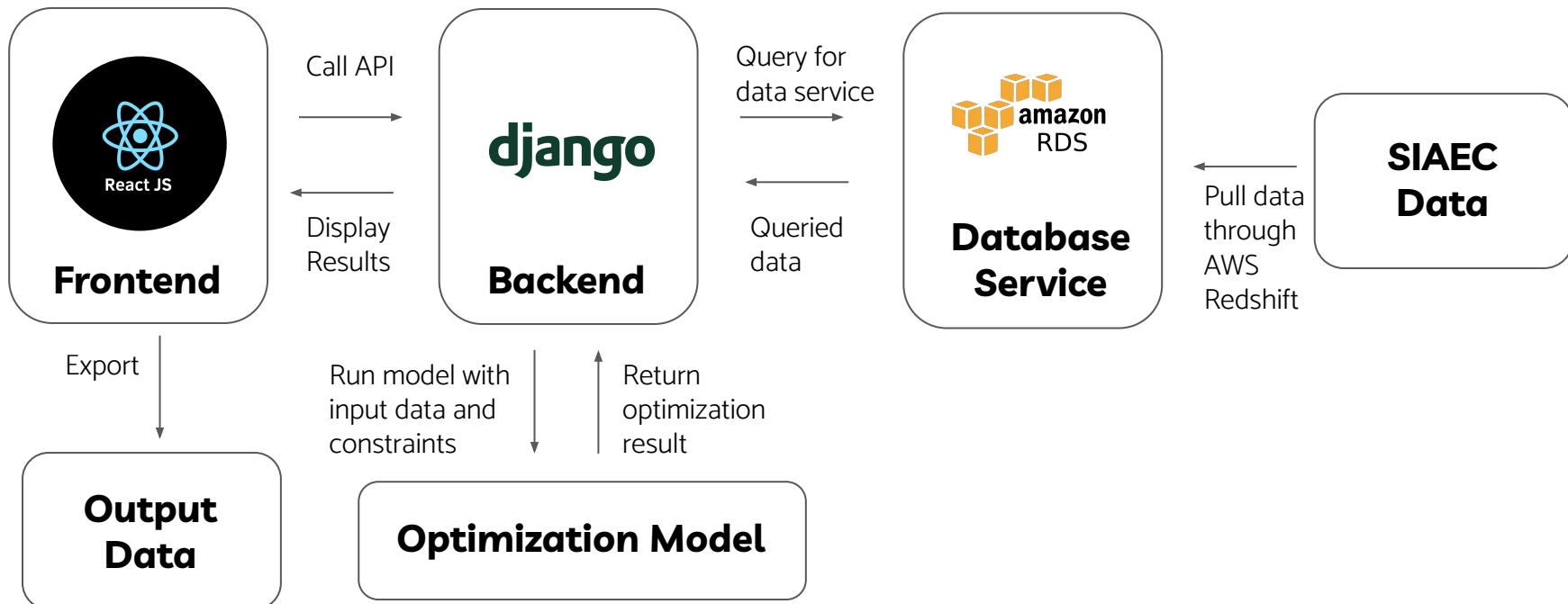
Database Service

Design Criteria	Alternative Design Concepts		
	Firebase	AWS RDS (Relational Database Service)	MongoDB Atlas
Performance	-	+	+
Applicability of Features	-	+	0
Cost-Efficiency	-	0	0
Total	---	++	+

* Further explanation of the Criteria used and assessment Rubrics can be find in Annex 1.6



Web Application Architecture Summary



Risk Assessment

Consequence Frequency Matrix
Potential Risk & Precautions
EHS

Consequence-frequency Matrix²²

		Impact		
		Small	Moderate	High
Frequency	Very High			
	High			
	Medium		Inaccurate Data Entry	Inaccurate Forecasting
	Low		Server Failure	Data Loss Security Breach

Potential Risk and Precautions

Event	Consequences	Mitigations
Inaccurate Forecasting	Potential increase in investment costs for SIAEC, leading to over provisioning	Writing test cases for algorithms
Inaccurate Data Entry	Inaccurate data displayed Decisions made based on incorrect data	Input validation
Security Breach	Stolen data Web application down	Follow the latest established security practices
Data Loss	Loss of valuable company data Web application unable to function	Data loss is hence mitigated by measures implemented by the cloud provider such as data replication
Server Failure	Web application would be down.	Will be using cloud service. Only physical disasters or virus attacks can result in a server failure



* More details can be found in Annex 1.7

Environment, Health & Safety

Safety	Sustainability
Security of Database	Ease of maintenance and update
Availability of service	Vulnerability of open source library
Confidentiality of user info	Flexibility to respond to future changes

Project Management

Limitations due to Covid-19
Project Task Allocation
Project Schedule
Budget Allocation

Limitations due to COVID-19

Limitations

- Easy to fall ill
- Shortened working hours



Solution

- Clear Objectives
- Delegation of tasks

Project Task Allocation

Responsibility Matrix (Exploration and Design Phase for Prototype)									R - Responsible	S - Supporting
	Task	Justin	Olivia	Kairan	Chester	YuYing	Devid	Wenqi		
1	UI/UX design	S	R							
2	Database Schema	R		R	S		S	R		
3	Data Cleaning	R		R				R		
4	Data Integration	R		R						
5	Operations Research				S	R	S			
6	Risk Management	R			R					
7	Project Management	R			R					
8	Updating of schedule	S			R					
9	Updates to budget		R					R		
10	Report	S	S	S	S	R	R	S		
11	Slides	S	R	S	S	S	S	S		
12	Exploration for frontend	S	R							
13	Exploration for database	R		S						
14	Exploration for backend	S		R						
15	Exploration for system architecture	S		R						

Project Schedule

March		April				May			
Concept and application selection									
	Wireframe for UI								
	Research theory and application								
	Formulation and completion of Just In Time Provisioning								
		Database Planning							
		Frontend and Backend Exploration							
			Review 2 Report and Slides						
			Just in Time Provisioning						
					Asset Management				
									Frontend and Backend Implementation, Testing
									Price Monitoring System

Project Schedule

June				July				August
Frontend and Backend Implementation Testing								
Price Monitoring System								
Review 3 slides								
Model Fine Tuning								
				Deploy Frontend and Backend				
					User Testing			
				Final Report and software documentation				
						Poster and summary of analysis result		
								Exhibition and Final Presentation

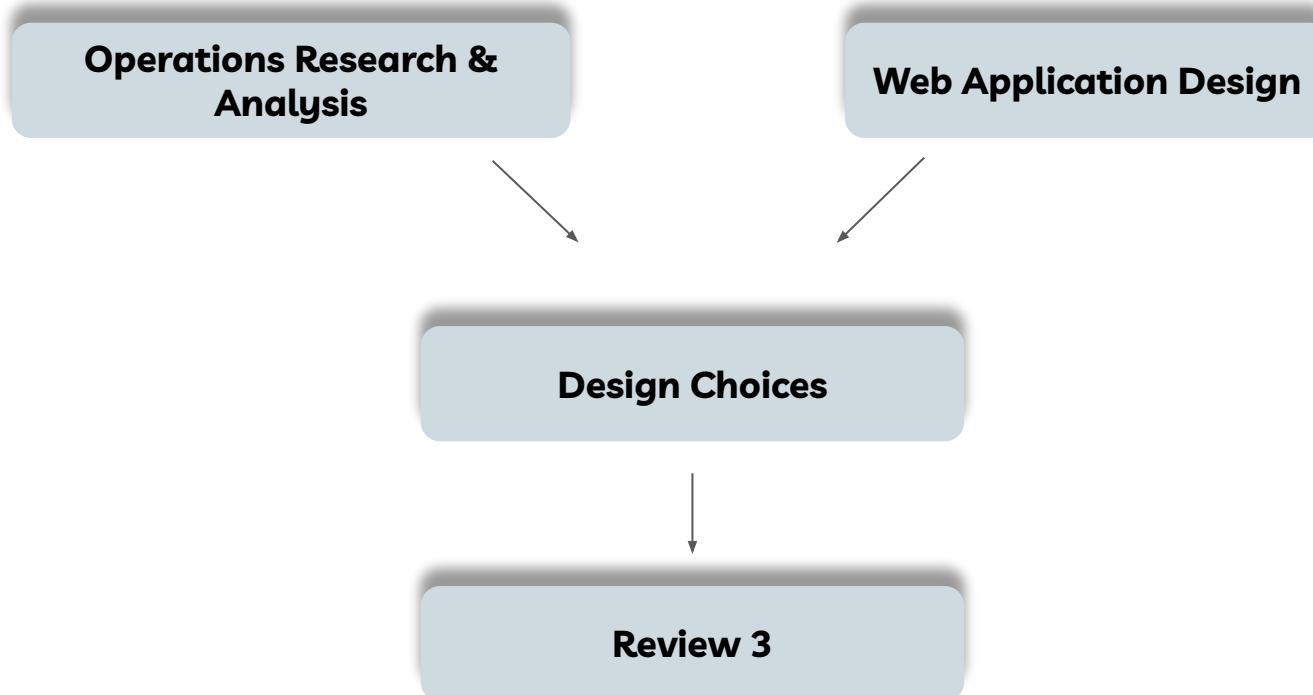
Expected Expenditure (SGD)	May	Jun	Jul	Aug
Tools				
Jira	52.50	52.50	52.50	52.50
Softwares				
RDS	200	200	200	200
EC2	287.50	287.50	287.50	287.50
Cloudfront	0	0	0	10
SSL Cert	0	0	0	25
Transportation				
Taxi	100	100	100	100
Total Expenditure	590	590	590	625

Budget Allocation

	May	Jun	Jul	Aug
Total Fund Available (SGD)	4,000	3,410	2,820	2,230
Total Expenditure (SGD)	590	590	590	625
Total Fund Remaining (SGD)	3,410	2,820	2,230	1,605

Conclusion

Conclusion



Q&A

Reference

References

Content

1. Our company profile. SIA Engineering Company. (n.d.). Retrieved March 2, 2022, from https://www.siaec.com.sg/company_profile.html
2. Ohri, A. (2017). 10 Popular Regression Algorithms In Machine Learning Of 2022. Retrieved 21 April 2022, from <https://www.jigsawacademy.com/popular-regression-algorithms-ml/#:~:text=Regression%20is%20a%20type%20of,the%20most%20popular%20regression%20algorithms>.
3. Wilkinson, G. (2020). Asset management optimization for repairable spare parts. Retrieved 21 April 2022, from <https://www.anylogic.com/blog/asset-management-optimization-for-repairable-spare-parts/>
4. Lye, K., & Chan, L. (2007). A Virtual Warehouse Simulation Tool for Aerospace Rotables Management. IEEE Aerospace Conference Proceedings.
5. Demand classification: Why forecastability matters - frepple APS. frePPLe. (2021, February 16). Retrieved April 12, 2022, from <https://frepple.com/blog/demand-classification/>
6. Chawla, G., & Miceli, V. (1970, January 1). Demand forecasting and inventory management for Spare Parts. Demand Forecasting and Inventory Management for Spare Parts. Retrieved April 20, 2022, from <https://dspace.mit.edu/handle/1721.1/121291>
7. Syntetos, A. A., Babai, M. Z., Lengu, D., & Altay, N. (1970, January 1). Distributional assumptions for parametric forecasting of intermittent demand. SpringerLink. Retrieved April 21, 2022, from https://link.springer.com/chapter/10.1007/978-0-85729-039-7_2#citeas
8. Pinçé, Ç., Turrini, L., & Meissner, J. (2021, July 3). Intermittent demand forecasting for spare parts: A critical review. Omega. Retrieved March 27, 2022, from <https://www.sciencedirect.com/science/article/pii/S0305048321001225>
9. Ofiwe, M. (2021, October 28). Metasearch Engines. SEMrush. Retrieved April 21, 2022, from <https://www.semrush.com/blog/metasearch-engine/>

References

Content

10. Wattanarat, V., Phimphavong, P., & Matsumaru, M. (2011). Demand and Price Forecasting Models for Strategic and Planning Decisions in a Supply Chain.
11. Why use serverless computing? | Pros and cons of serverless. (n.d.). Cloudflare. Retrieved April 22, 2022, from <https://www.cloudflare.com/en-gb/learning/serverless/why-use-serverless/>
12. Lange, K. (2021, February 25). What's Serverless? Pros, Cons & How Serverless Computing Works. BMC Software. Retrieved April 22, 2022, from <https://www.bmc.com/blogs/serverless-computing/>
13. Bakusevych, T. (2021, March 2). *Top vs side navigation: Which one is better for your product?* UX Collective. Retrieved April 22, 2022, from <https://uxdesign.cc/top-navigation-vs-side-navigation-wich-one-is-better-24aa5d835643>
14. Borrelli, P. (2021, October 26). Angular vs. React vs. Vue.js: Comparing performance. LogRocket Blog. Retrieved April 21, 2022, from <https://blog.logrocket.com/angular-vs-react-vs-vue-js-comparing-performance/>
15. Jonah, V. (2021). 5 top Go web frameworks - LogRocket Blog. Retrieved 21 April 2022, from <https://blog.logrocket.com/5-top-go-web-frameworks/>
16. Volodymyr, T. (2017). Express.js Mobile App Development: pros and cons of Node.js framework. Retrieved 21 April 2022, from <https://apiko.com/blog/express-mobile-app-development/>
17. Bhatt, S. (2020). Pros and Cons of Django Framework for App Development - DZone Web Dev. Retrieved 21 April 2022, from <https://dzone.com/articles/pros-and-cons-of-django-framework-for-app-developm>
18. Gin Web Framework. Retrieved 21 April 2022, from <https://gin-gonic.com/>
19. DataFlair,. Retrieved 21 April 2022, from <https://data-flair.training/blogs/django-advantages-and-disadvantages/>
20. Wodehouse, C. (2018). Firebase vs. AWS vs. MongoDB: 3 Technologies Behind Modern Software Stacks. Retrieved 21 April 2022, from <https://www.business2community.com/brandviews/upwork/firebase-vs-aws-vs-mongodb-3-technologies-behind-modern-software-stacks-02147649>

Content

21. Singh, M. (2021). Social Network for Programmers and Developers. Retrieved 21 April 2022, from <https://morioh.com/p/fd327efa3b32>
22. van Westen, C. (n.d.). 5.5 Risk assessment methods. Caribbean Disaster Emergency Management Agency - CDEMA. Retrieved April 20, 2022, from
<https://www.cdema.org/virtuallibrary/index.php/charim-hbook/methodology/5-risk-assessment/5-5-risk-assessment-methods>

References

Design Credit

Presentation template by [Slidesgo](#)

Icons by [Flaticon](#)

Infographics by [Freepik](#)

Images

<https://www.simplilearn.com/data-analysis-methods-process-types-article>

https://www.123rf.com/photo_36638346_modern-notebook-computer-with-future-technology-media-symbols.html

Annex

Short-term Provisioning

Criteria\Score	-	0	+
Data Requirement	The method requires a few years of historical data	The method requires a few months a historical data	The method requires a month worth of historical data
Accuracy	The method produces poor accuracy result	The method produces acceptable accuracy result	The method produces exceptional accuracy result
Time	The method requires a lot of time to implement	The method uses moderate amount of time to implement	The method requires very little time to implement
Ease of implementation	The method is computationally intensive with many mathematical equations and concepts	The method is moderately easy to implement with fair amount of mathematical equations and concepts	The method is easy to implement with little mathematical equations and concepts

Annex 1.2

Price Monitoring System (Price Acquisition)

Criteria\Score	-	0	+
Affordability	Software costs more than SGD 100	Software costs less than SGD 100	This software is free to use
Ease of Integration	Software is difficult to integrate relative to the other alternatives	Software is reasonably difficult to integrate relative to the other alternatives	Software is easy to integrate relative to the other alternatives
Adaptability	Requires major tuning whenever website is changed relative to the other alternatives	Requires moderate tuning whenever website is changed relative to the other alternatives	Requires little to no tuning whenever website is changed relative to the other alternatives

Price Monitoring System (Price Forecasting)

Criteria\Score	-	0	+
Data Requirement	The method requires a few years of historical data	The method requires a few months a historical data	The method requires a month worth of historical data
Accuracy	The method produces poor accuracy result	The method produces acceptable accuracy result	The method produces exceptional accuracy result
Time	The method requires a lot of time to implement	The method uses moderate amount of time to implement	The method requires very little time to implement
Ease of implementation	The method is computationally intensive with many mathematical equations and concepts	The method is moderately easy to implement with fair amount of mathematical equations and concepts	The method is easy to implement with little mathematical equations and concepts

Annex 1.4

Frontend Framework

Criteria\Score	-	0	+
Ease of Learning	All team members have no prior knowledge with the framework	Some team members have some prior knowledge with the framework	Every members have prior experience with the framework
Performance	Has slow rendering speed, slower response speed with increased number of features relative to the other alternatives	Has a moderate rendering speed, no significant reduction in response speed with increased number of features relative to the other alternatives	Has fast rendering speed, no significant reduction in response speed with increased number of features relative to the other alternatives
Applicability of Features	Offers few relevant features fulfilling the needs and constraint of the project relative to the other alternatives	Offers some relevant features fulfilling the needs and constraint of the project relative to the other alternatives	Offers many relevant and useful features fulfilling the needs and constraint of the project relative to the other alternatives

Annex 1.5

Backend Framework

Criteria\Score	-	0	+
Ease of Learning	All team members have no experience with the framework, few learning resources available	All team members have no experience with the framework, some learning resources available	Some team members have experience with the framework, some learning resources available
Ease of Implementation	Framework requires a lot of code, rigid in implementation	Framework requires a moderate amount of code, relatively flexible in implementation	Framework requires a moderate amount of code, flexible in implementation
Performance	Slow response time relative to the other alternatives	Moderate response time relative to the other alternatives	Fast response time relative to the other alternatives

Annex 1.6

Database Service

Criteria\Score	-	0	+
Performance	Slow reads and writes with poor concurrency capabilities relative to the other alternatives	Fast reads and writes with moderate concurrency capabilities relative to the other alternatives	Fast reads and writes with good concurrency capabilities relative to the other alternatives
Applicability of Features	Has few relevant features offered by service provider relative to the other alternatives	Has some relevant features offered by service provider relative to the other alternatives	Has many relevant and useful features offered by service provider relative to the other alternatives
Cost-Efficiency	Monthly expenditure is above SGD 300	Monthly expenditure is between SGD 300 and SGD 100	Monthly expenditure is below SGD 100

Potential Risk and Precautions

Event	Description	Consequences	Mitigations
Inaccurate Forecasting	Caused by the algorithm's inaccuracy to forecast out the optimal number of parts for SIAEC to provision or to keep as active stock	Potential increase in investment costs for SIAEC, leading to over provisioning	Writing test cases for algorithms
Inaccurate Data Entry	Data input into the database is wrong	Inaccurate data displayed Decisions made based on incorrect data	Input validation
Security Breach	Caused by human error and loopholes in software.	Stolen data Web application down	Follow the latest established security practices
Data Loss	Caused by human error, viruses, natural disasters, power failure	Loss of valuable company data Web application unable to function	Data loss is hence mitigated by measures implemented by the cloud provider such as data replication
Server Failure	Caused by disk failure, virus attack, failed updates and physical disasters	Web application would be down.	Will be using cloud service. Only physical disasters or virus attacks can result in a server failure