

Burger Specialist

- the architecture of a web social platform

Jiahui TANG

2023/03/16

Content

- General Design
 - Simple Specification
 - System Context
 - Containers
 - Data Security & User Privacy Solution
 - Increased Workloads Solution
- Further Thoughts and Design
- Demo Detail

Simple Specification

- **A new social platform**

- Review local burger places

Store the information of burger places

- Score the burger places

User can make a comment on the burger places

- Upload pictures

Contain file servers to save pictures or other media data

- Find nearby restaurants

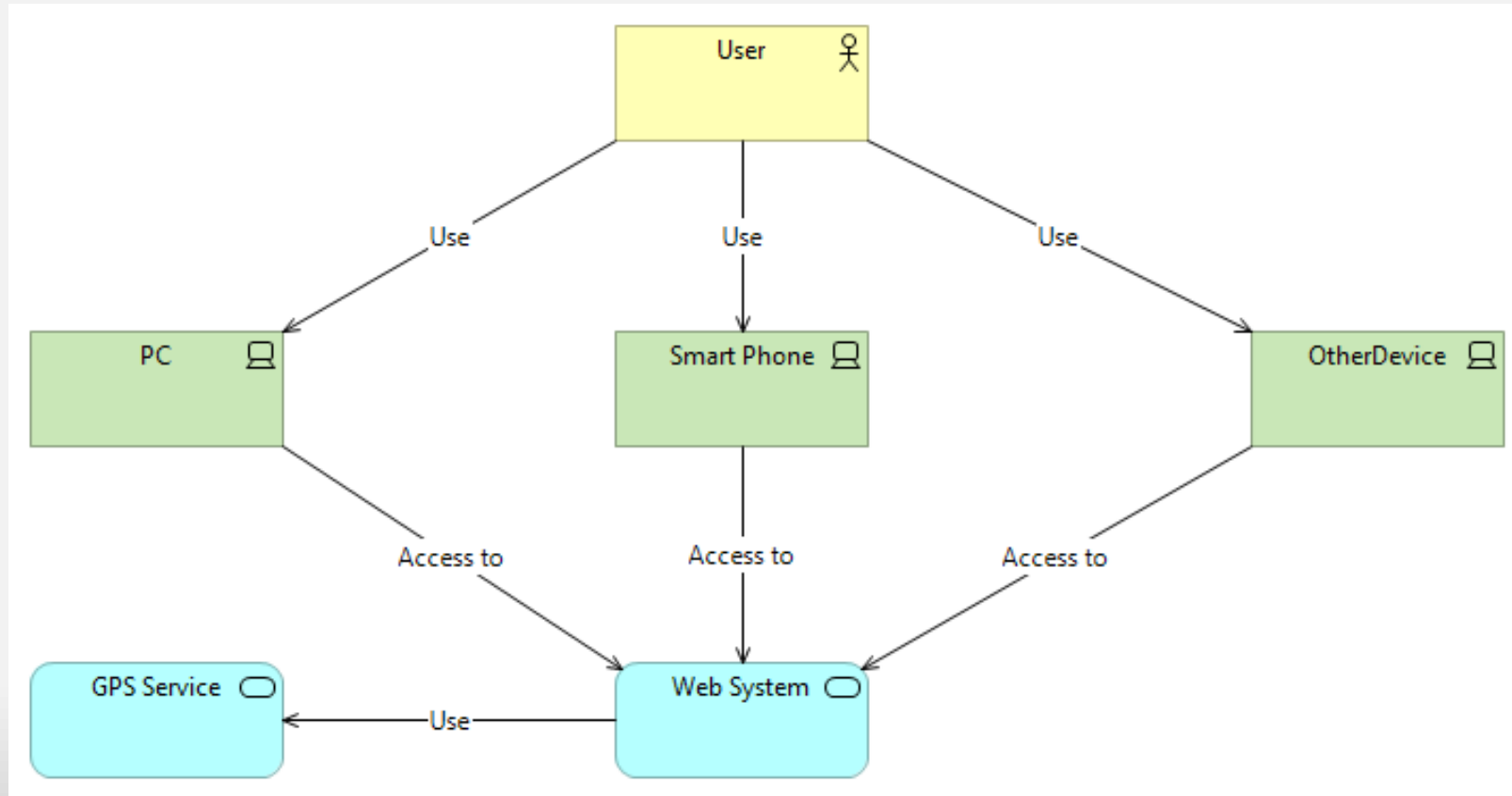
Find restaurants according to a confirmed location



- **Available on multi-platform**

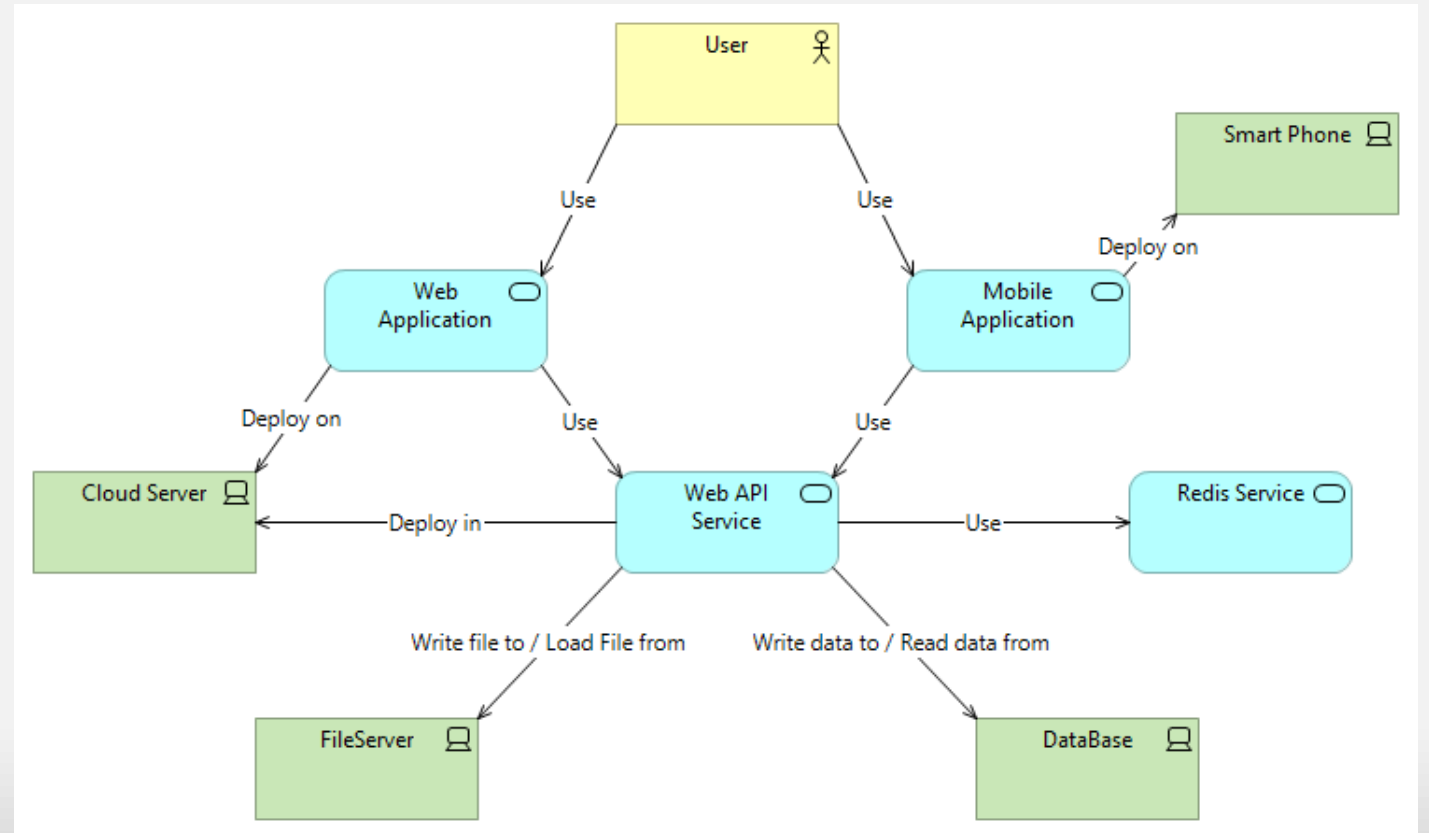
- ✓ A Web application
- ✓ Should be able to be deployed on multi-platform
- ✓ Need a file system to save pictures and videos
- ✓ Need a database to save user information and other records
- ✓ **Data security and user privacy should be concerned**
- ✓ **The system should be able to handle the increased workloads**

System Context



Container

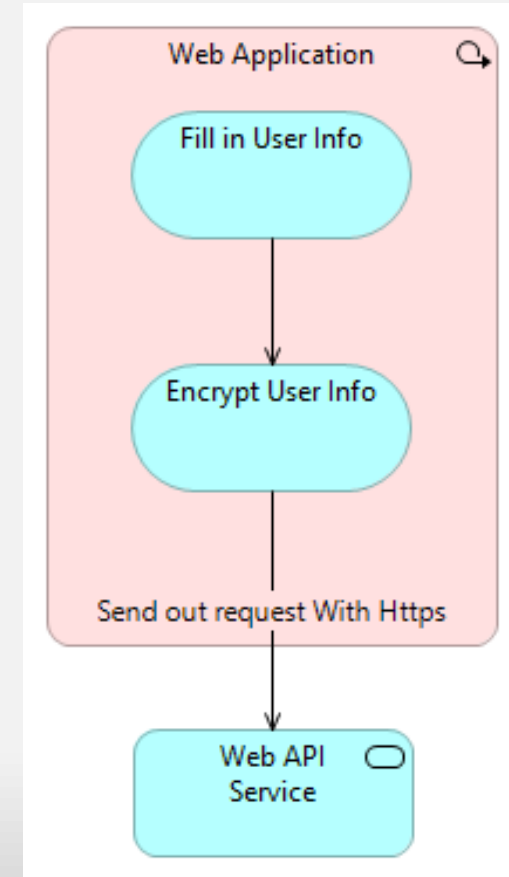
- Three Container:
 - Web application
 - Mobile Application
 - Asp .NET Application
- Multi-Platform:
 - Ionic + Cordova + Web application



Data Security & User Privacy

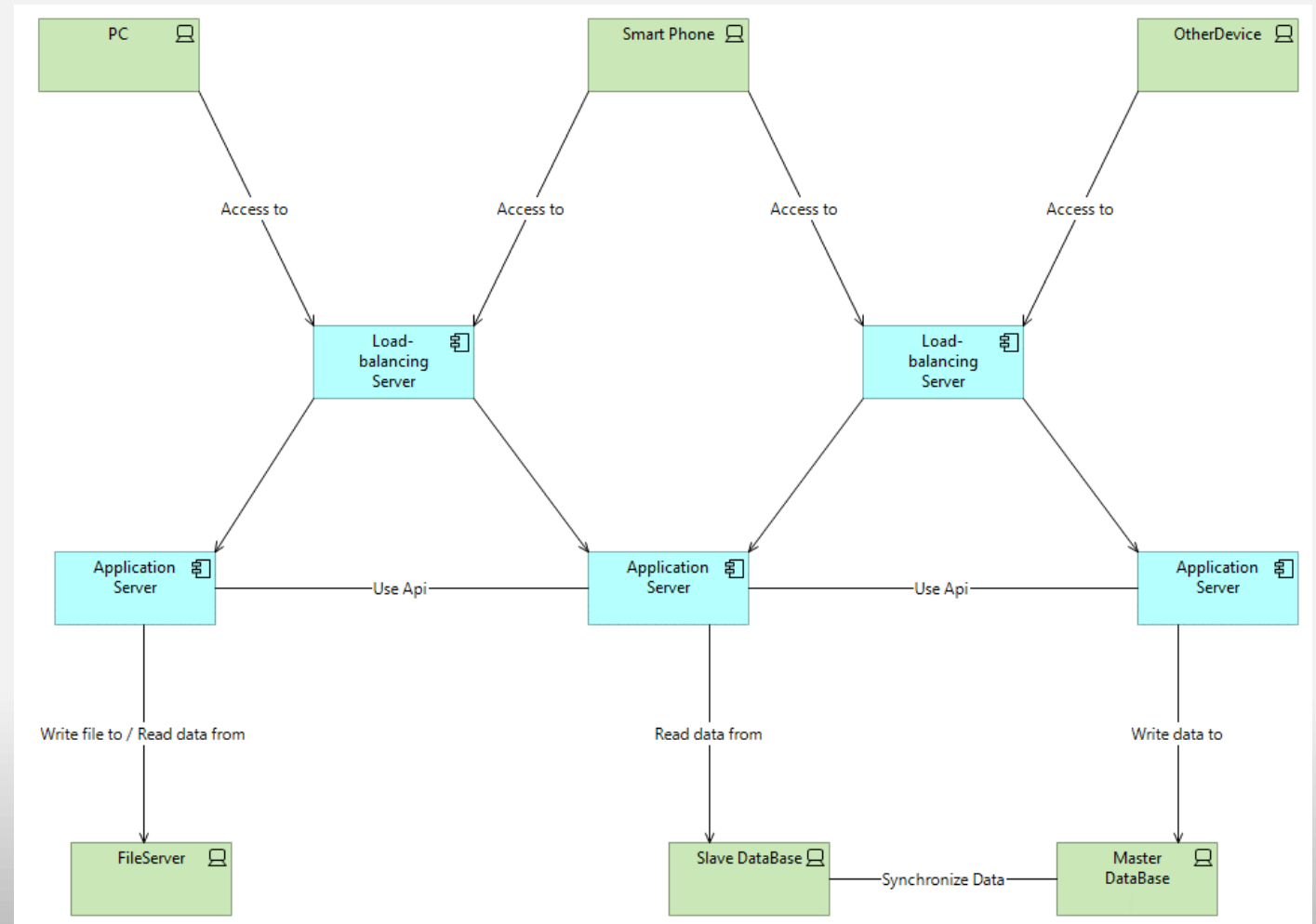
Use Https protocol and pre-encryption process to ensure the data security and User privacy

- Https:
Https protocol ensures the data security when transmitting request between servers
- Pre-encryption process:
Encrypting the user info (name, password, etc.) before sending out the https request ensures all the user data could be saved in the system in the encrypted state and can only be decrypted by our own system.

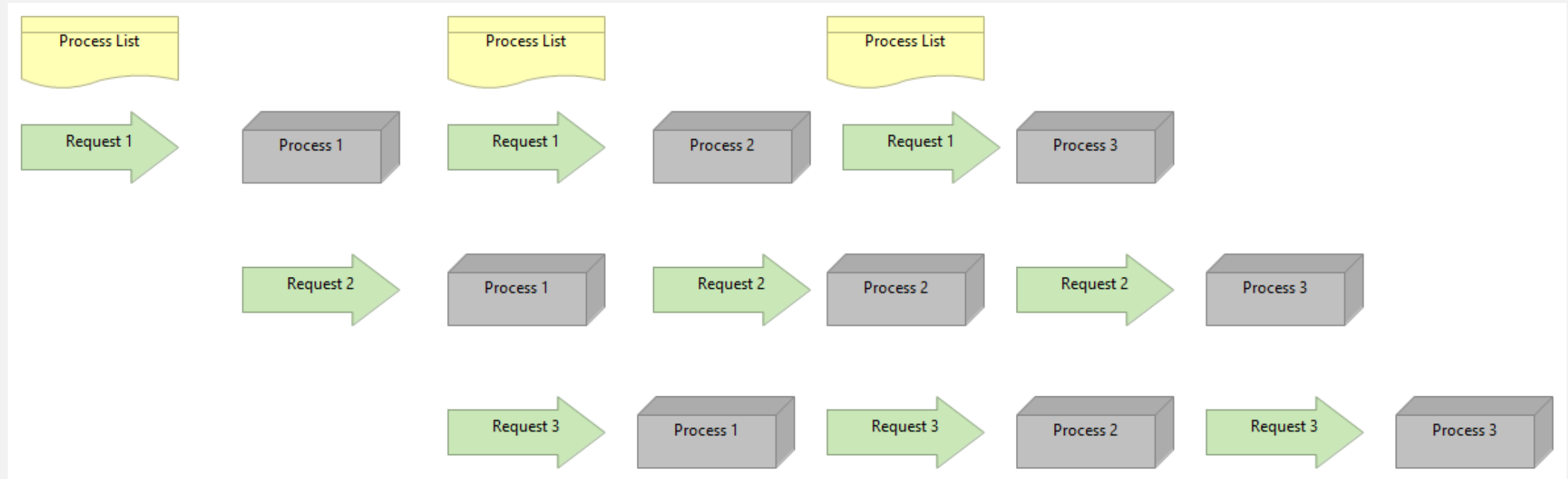


Increased workloads

- Bottlenecks:
 - Server's load
 - I/O Bottleneck
 - DB Bottleneck
- Solutions:
 - ✓ Use load-balancing server
 - ✓ Use Redis service
 - ✓ Use master-slave database
 - ✓ Partition the db. tables



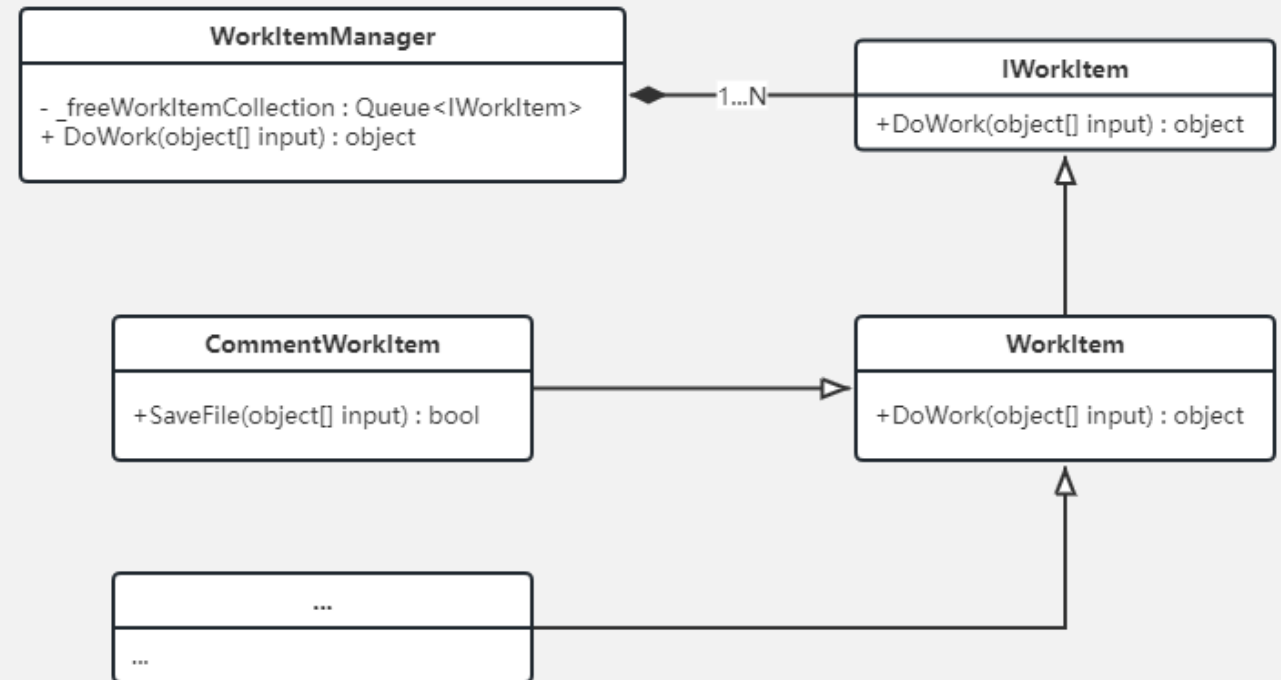
Further Design – pipeline mode



- A function would be separate into many process node, each node should cost almost the same amount of time.
- After a process node is finished, re-query a free node resource to execute next process.
- The API behavior could be totally described by a config file.

Detail Info of Demo

- WorkItemManager: Manage the workitem resource of the system
- IWorkItem: The Interface which defines the behavior of WorkItem
- WorkItem: The abstract implement class of IWorkItem
- CommentWorkItem: The actual class managed by the WorkItemManager



Thank you for your watching!

Q&A