



Software and Data Architecture for ReClothes

Terrier

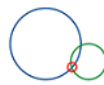
Table of Contents

OVERVIEW	3
SOFTWARE ARCHITECTURE	8
High-Level logical components and their interactions	8
Description of the Modules or system components of the system	9
Outline the Type of connections and integration patterns	15
Architecture styles used	17
Data Architecture	20
Identify Data Entities for each User Story	20
Build data architecture	24

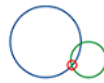
1. OVERVIEW

User Stories	Quality attributes mapping	Justification for the mapping
Epic 01: Common Functions		
#c-01 #c-04 External User Registration	Security	Security - The system must keep all external users' information secure and keep restricted access permission for the different internal user roles. For example, all passwords must be hashed with salt. And sensitive user information can't be accessed by common internal employees.
#c-02 #c-05 External User Login	Security	Security - The system can use CAPTCHA to prevent bots from login into the system. And the system can provide some OAuth features to let users use third-party accounts to log in.
#c-03 #c-06 External User Logout	Security	Security - Remove the session data related to the user.
#c-07 #c-09 #c-11 Internal User Login	Security	Security - The system must restrict the employees can only log in to the system from their internal network, they can't access the system from the internet.
#c-08 #c-10 #c-12 Internal User Logout	Security	Security - Remove the session data related to the user.
Epic 02: Forward Donating Process		
#d-pr-01 #d-pr-02 #d-pr-03	Usability	Usability - Easy and simple donating process can give users more satisfaction and let them keep the habit of

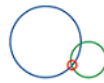
#d-pr-04 Pre-Donating		sustainability.
#d-do-01 #d-do-02 #d-do-03 Donating	Usability Fault Tolerance	Usability - Donators can use their phone to open the drop-off box easily Fault Tolerance - As all drop-off boxes must connect with the internet to operate correctly, these drop-off boxes should still run reliably after a network outage.
#d-co-01 #d-co-02 #d-co-3 #d-co-4 Collecting	Portability	Portability - Deliveryman can use their phone to run our system and also use their phone to connect with their portable printer for printing out tracking code.
#d-po-01 #d-po-02 #d-po-3 #d-po-4 #d-po-5 #d-po-6 #d-po-7 #d-po-8 #d-po-9 #d-po-10 Post-Donating	Supportability	Supportability - All operations made by employees must be recorded persistently in our system for troubleshooting.
Epic 03: Forward Buying Process		
#b-pr-01 #b-pr-02 Pre-Buying	Availability Performance	Availability - The system should keep available when buyers add clothes into their shopping cart and keep the shopping process smoothly. Performance - The system should respond to buyers as quickly as possible as each delay in the shopping process will lose buyers' patience.
#b-bu-01 #b-bu-02 #b-bu-03 #b-bu-04 #b-bu-05	Security Fault Tolerance	Security - As this process contains submitting the orders and paying the order, the security among this process becomes more important than before. The system should keep buyers' credit



Buying		card information securely. Fault Tolerance - The system should connect with some third-party payment tools such as Stripe or Paypal, so the system should keep running correctly after occurring some connection issues with the third-party payment platform.
#b-pa-01 #b-pa-02 #b-pa-03 Packing	Supportability	Supportability - All operations made by employees must be recorded persistently in our system for troubleshooting.
#b-sh-01 #b-sh-02 Shipping	Portability	Portability - Deliveryman can use their phone to run our system and also use their phone to connect with their portable printer for printing out tracking codes.
#b-po-01 #b-po-02 #b-po-03 #b-po-04 Post-Buying	Fault Tolerance	Fault Tolerance - As all drop-off boxes must connect with the internet to operate correctly, these drop-off boxes should still run reliably after a network outage.
Epic 04: Reverse Donating Process		
#rd-c-01 #rd-c-02 #rd-c-03 #rd-c-04 Cancel Donating Order	Usability	Usability - The system should give donators a consistent feeling of smoothness not only in the process of donating but also in the process of canceling or returning.
#rd-r-re-01 #rd-r-re-02 Returning	Usability	Usability - The system should give donators a consistent feeling of smoothness not only in the process of donating but also in the process of canceling or returning.
#b-pa-01 #b-pa-02 #rd-r-pa-01 Packing	Supportability	Supportability - All operations made by employees must be recorded persistently in our system for troubleshooting.
#b-sh-01	Portability	Portability - Deliveryman can use their phone to run our system and also use



#b-sh-02 Shipping		their phone to connect with their portable printer for printing out tracking codes.
#rd-r-po-01 #rd-r-po-02 #rd-r-po-03 #rd-r-po-04 Post Returning	Fault Tolerance	Fault Tolerance - As all drop-off boxes must connect with the internet to operate correctly, these drop-off boxes should still run reliably after a network outage.
Epic 05: Reverse Buying Process		
#rb-c-01 #rb-c-02 Cancel Buying Order	Usability	Usability - The system should give buyers a consistent feeling of smoothness not only in the process of buying but also in the process of canceling or returning.
#rb-r-re-01 #rb-r-re-02 #rb-r-re-03 Returning	Usability	Usability - The system should give buyers a consistent feeling of smoothness not only in the process of buying but also in the process of canceling or returning.
#d-co-01 #d-co-02 #d-co-02 #rb-r-co-01 Collecting	Portability	Portability - Deliveryman can use their phone to run our system and also use their phone to connect with their portable printer for printing out tracking codes.
#d-po-01 #d-po-02 #rb-r-po-01 #rb-r-po-02 #rb-r-po-03 #rb-r-po-04 #d-po-07 #d-po-08 Post Returning	Supportability	Supportability - All operations made by employees must be recorded persistently in our system for troubleshooting.
Epic 06: Management Functions		
#mf-eu-01 #mf-eu-02 #mf-eu-03	Security Legality	Security - Not all employees can check the detailed information of buyers or buyers and some sensitive information

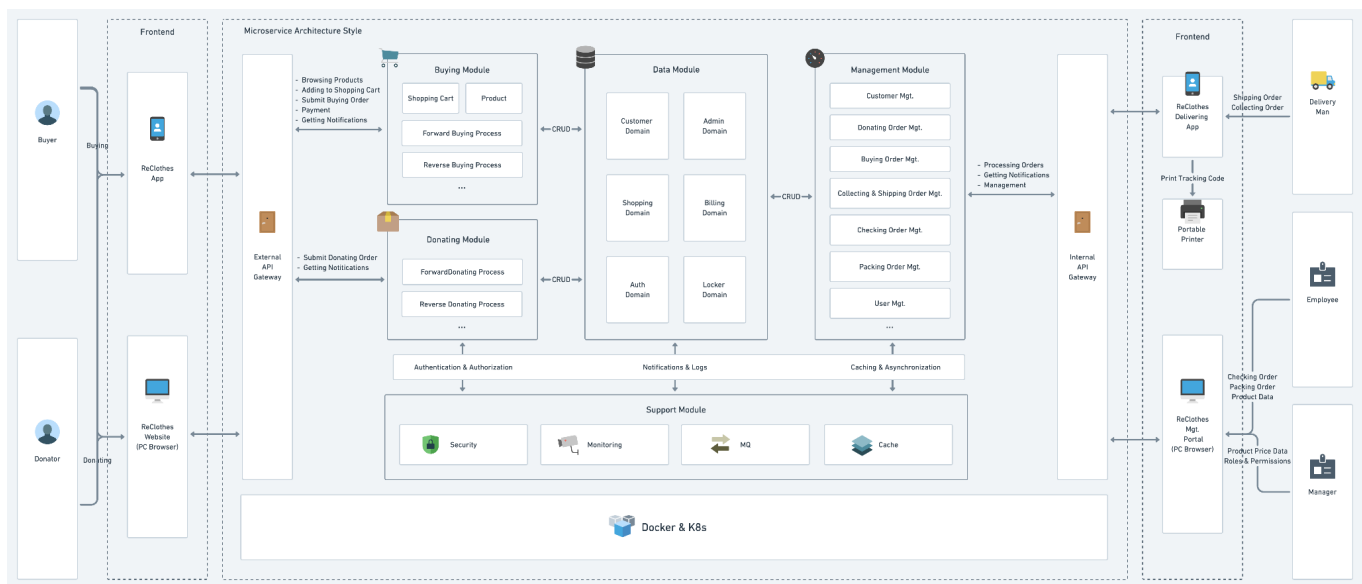


#mf-eu-04 #mf-eu-05 #mf-eu-06 External User Mgt.		cannot show to anyone. Legality - The system should follow the legislative constraints of data protection or GDPR.
#mf-iu-01 #mf-iu-02 Internal User Mgt.	Supportability	Supportability - The system should keep logging all operations made by internal users.
#mf-bo-01 #mf-bo-02 #mf-bo-03 Buying Order Mgt. #mf-do-01 #mf-do-02 #mf-do-03 Donating Order Mgt. #mf-so-01 #mf-so-02 #mf-so-03 #mf-so-04 Shipping Order Mgt. #mf-co-01 #mf-co-02 #mf-co-03 #mf-co-04 #mf-co-05 Collecting Order Mgt. #mf-po-01 #mf-po-02 #mf-po-03 #mf-po-04 Packing Order Mgt. #mf-cho-01 #mf-cho-02 #mf-cho-03 Checking Order Mgt.	Authorization Supportability	Authorization - The system should keep different roles can only access partial information that needed be processed by this role. And only the super admin or managers can access all information. Supportability - The system should keep logging all operations made by internal users.
#mf-op-01 #mf-op-02 #mf-op-03 #mf-op-04	Supportability	Supportability - The system should keep logging all operations made by internal users. And the system should pay more attention to the changes in products'

#mf-op-05 Online Product Mgt.		price.
#mf-dob-01 #mf-dob-02 #mf-dob-03 #mf-dob-04 Drop-off Box Mgt.	Supportability	Supportability - The system should keep logging all operations made by internal users.
#mf-oc-01 #mf-oc-02 #mf-oc-03 Off-line Coupon Mgt.	Supportability	Supportability - The system should keep logging all operations made by internal users. And the system should keep recording the usage of each offline coupon.

1 SOFTWARE ARCHITECTURE

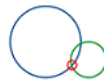
1.1 HIGH-LEVEL LOGICAL COMPONENTS AND THEIR INTERACTIONS



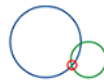
digital version: <https://whimsical.com/tech-architecture-H6WEoQvhq5RPDWMQHDZo5Z>

1.2 DESCRIPTION OF THE MODULES OR SYSTEM COMPONENTS OF THE SYSTEM

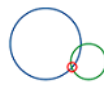
Component Name	Description	User Stories associated
ReClothes App	<p>This mobile app provides all functions related to buyers and donators.</p> <p>Buyers can use this app to browse all kinds of used clothes, then add one or two into their shopping cart. In the shopping cart, buyers can see the total price and choose whether or not to create a buying order. After creating the buying order, buyers must choose their address information and make a payment. After several days, buyers can use this app to scan the QR code on the drop-off box to open the door, then grab out clothes and get some AR animations or sustainability coins for stimulation.</p> <p>Donators can use this app to type into the information of their used clothes into our system, then they can use this app to open the door of the drop-off box. After closing the door, they also can get some AR animations for stimulation. As soon as their used clothes are finished the checking process, their clothes are either put into the online store to sell or send to some third-party factories for recycling. And they can get some offline coupons or sustainability coins for stimulation.</p>	<p>epic-01: #c-01, #c-02, #c-03, #c-04, #c-05, #c-06, #c-011, #c-012</p> <p>epic-02: #d-pr-01, #d-pr-02, #d-pr-03, #d-pr-04, #d-do-01, #d-do-02, #d-do-03, #d-po-03, #d-po-04, #d-po-09, #d-po-10</p> <p>epic-03: #b-pr-01, #b-pr-02, #b-bu-01, #b-bu-02, #b-bu-03, #b-bu-04, #b-bu-05, #b-po-01, #b-po-02, #b-po-03, #b-po-04</p> <p>epic-04: #rd-c-01, #rd-c-02, #rd-c-03, #rd-c-04, #rd-r-re-01, #rd-r-re-02, #rd-r-pa-01, #rd-r-po-01, #rd-r-po-02, #rd-r-po-03, #rd-r-po-04</p> <p>epic-05: #rb-c-01, #rb-c-02, #rb-r-re-01, #rb-r-re-02, #rb-r-re-03, #rb-r-co-01, #rb-r-po-01, #rb-r-po-03</p>
ReClothes Website	<p>Some functions are shown above in the mobile app. But the UI design is more convenient for the PC browser.</p> <p>Buyers and donators can get more detailed information on our website. And the app only provides essential and core functions.</p>	<p>epic-01: #c-01, #c-02, #c-03, #c-04, #c-05, #c-06, #c-011, #c-012</p> <p>epic-02: #d-pr-01, #d-pr-02, #d-pr-03, #d-pr-04, #d-do-01, #d-do-02, #d-do-03, #d-po-03, #d-po-04, #d-po-09, #d-po-10</p> <p>epic-03: #b-pr-01, #b-pr-02, #b-bu-01, #b-bu-02, #b-bu-03,</p>



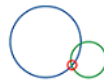
		<p>#b-bu-04, #b-bu-05, #b-po-01, #b-po-02, #b-po-03, #b-po-04</p> <p>epic-04: #rd-c-01, #rd-c-02, #rd-c-03, #rd-c-04, #rd-r-re-01, #rd-r-re-02, #rd-r-pa-01, #rd-r-po-01, #rd-r-po-02, #rd-r-po-03, #rd-r-po-04</p> <p>epic-05: #rb-c-01, #rb-c-02, #rb-r-re-01, #rb-r-re-02, #rb-r-re-03, #rb-r-co-01, #rb-r-po-01, #rb-r-po-03</p>
ReClothes Delivering App	<p>This app is only provided for delivery men.</p> <p>As almost all tasks of delivery men need to finish outside of the company, the management website may be convenient for them to use. In addition, they must use this app to connect to their portable printer to print out the tracking code stickers.</p> <p>In this app, delivery men can easily get notifications of new collecting or shipping tasks. The collecting task is the direction from drop-off locations to the company, and the shipping task is the reverse direction from the company to drop-off locations.</p> <p>After they collected the used clothes from drop-off boxes, they can click the “print” button on the app to let the portable printer to print out a tracking code sticker to stick on top of the package for subsequent tracing.</p> <p>BTW, for higher security, this app can only access from a VPN. Only users who log in successfully into the VPN can access this app.</p>	<p>epic-01: #c-11, #c-12</p> <p>epic-02: #d-co-01, #d-co-02, #d-co-03</p> <p>epic-03: #b-sh-01, #b-sh-02</p> <p>epic-06: #mf-so-01, #mf-so-02, #mf-so-03, #mf-so-04, #mf-co-01, #mf-co-02, #mf-co-03, #mf-co-04, #mf-co-05</p>
Portable Printer	<p>It’s a portable and easy-connected mini printer. The delivery man can use their mobile phone to connect with the</p>	<p>epic-02: #d-co-03</p>



	ReClothes Delivering app for printing tracking codes.	
ReClothes Mgt. Portal	<p>Employees and managers can use this portal to manage all kinds of information. Such as user management, customer management, buying order management, donating order management, etc.</p> <p>Employees can only check information related to them, not all information they can access. The permissions must be assigned to them by their managers.</p> <p>Managers have higher permissions to check more information and they can assign different checking orders, packing orders, collecting orders, shipping orders to different employees or delivery men. For more fine-grained management, there's a user management role that can give internal users permissions to activate or deactivate users, assign different roles to different internal users, or some other advanced management functions.</p> <p>BTW, internal users can only access this website from the internal network or a VPN in the outside network.</p>	<p>epic-01: #c-07, #c-08, #c-09, #c-10</p> <p>epic-02: #d-po-01, #d-po-02, #d-po-05, #d-po-06, #d-po-07, #d-po-08</p> <p>epic-03: #b-pa-01, #b-pa-02</p> <p>epic-05: #rb-r-po-02, #rb-r-po-04</p> <p>epic-06: #mf-eu-01, #mf-eu-02, #mf-eu-03, #mf-eu-04, #mf-eu-05, #mf-eu-06, #mf-iu-01, #mf-iu-02, #mf-bo-01, #mf-bo-02, #mf-bo-03, #mf-do-01, #mf-do-02, #mf-do-03, #mf-so-01, #mf-so-02, #mf-so-03, #mf-so-04, #mf-co-01, #mf-co-02, #mf-co-03, #mf-co-04, #mf-co-05, #mf-po-01, #mf-po-02, #mf-po-03, #mf-po-04, #mf-cho-01, #mf-cho-02, #mf-cho-03, #mf-op-01, #mf-op-02, #mf-op-03, #mf-op-04, #mf-op-05, #mf-dob-01, #mf-dob-02, #mf-dob-03, #mf-dob-04, #mf-oc-01, #mf-oc-02, #mf-oc-03</p>
External API Gateway	It's a gateway only for external users (buyers and donators). After their requests reached this gateway, the gateway can do some authentications, log activities, transfer their requests to a specific service for subsequent processing.	All user stories are related to buyers and donators



Internal API Gateway	<p>This API gateway only works for receiving requests from internal users (employees, delivery men, managers). Users can only send requests to this gateway from the company network or use a VPN from the outside network. Just like the external API gateway, this gateway can also provide authentication, logging function, and transfer requests to a specific service for further processing.</p>	<p>All user stories are related to delivery men, employees, and managers.</p>
Buying Module	<p>Buying module is a distributed system that consists of several different services, such as customer service, shopping cart service, product service, finance service, external order service, and locker service. These services act like a complete system that can give buyers features for the forward buying process and the reverse buying process.</p> <p>Since we consider using microservice to build the whole system, the connection between these different services is HTTP connection. Each request between different services is combined with an access token for authentication.</p>	<p>epic-03: #b-pr-01, #b-pr-02, #b-bu-01, #b-bu-02, #b-bu-03, #b-bu-04, #b-bu-05, #b-pa-03, #b-po-01, #b-po-02, #b-po-03, #b-po-04</p> <p>epic-04: #rd-r-po-04</p> <p>epic-05: #rb-c-01, #rb-c-02, #rb-r-re-01, #rb-r-re-02, #rb-r-re-03, #rb-r-co-01, #rb-r-po-01, #rb-r-po-03</p>
Donating Module	<p>Same with the donating module, this is also a distributed system that combined customer service, finance service, external order service, and locker service. These services act like a complete system that can give donators features for the forward donating process and the reverse donating process.</p> <p>Same with the buying module, connections between these services are HTTP requests with an access token for authentication.</p>	<p>epic-02: #d-pr-01, #d-pr-02, #d-pr-03, #d-pr-04, #d-do-01, #d-do-02, #d-do-03, #d-co-04, #d-po-03, #d-po-04, #d-po-09, #d-po-10</p> <p>epic-04: #rd-c-01, #rd-c-02, #rd-c-03, #rd-r-re-01, #rd-r-re-02, #rd-r-pa-01, #rd-r-po-01, #rd-r-po-02, #rd-r-po-03</p>
Management Module	<p>Same with the previous two modules, the management module is also a distributed system that consists of all services (customer service, product service, shopping cart service, finance service, external order service, locker service,</p>	<p>epic-06: #mf-eu-01, #mf-eu-02, #mf-eu-03, #mf-eu-04, #mf-eu-05, #mf-eu-06, #mf-iu-01, #mf-iu-02, #mf-bo-01, #mf-bo-02, #mf-bo-03, #mf-do-01,</p>



	<p>internal order service, and admin service). This module gives internal users all kinds of management functions. Through this module, internal users can check the status and progress of a specific order. Or managing internal users and external users.</p> <p>Same with the previous two modules, connections between these services are HTTP connections with an access token for authentication.</p>	<p>#mf-do-02, #mf-do-03, #mf-so-01, #mf-so-02, #mf-so-03, #mf-so-04, #mf-co-01, #mf-co-02, #mf-co-03, #mf-co-04, #mf-co-05, #mf-po-01, #mf-po-02, #mf-po-03, #mf-po-04, #mf-cho-01, #mf-cho-02, #mf-cho-03, #mf-op-01, #mf-op-02, #mf-op-03, #mf-op-04, #mf-op-05, #mf-dob-01, #mf-dob-02, #mf-dob-03, #mf-dob-04, #mf-oc-01, #mf-oc-02, #mf-oc-03</p>
Data Module	<p>Since we use microservices architecture, for avoiding all kinds of coupling, including shared schemas and databases, we split the database into eight sub-databases.</p> <ul style="list-style-type: none">- Customer Data- Product Data- Shopping Cart Data- External Order Data<ul style="list-style-type: none">- the buying order- the donating order- Admin Data- Finance Data<ul style="list-style-type: none">-- Locker Data<ul style="list-style-type: none">- drop-off box- drop-off location- Internal Order Data<ul style="list-style-type: none">- the checking order- the packing order- the collecting order- the shipping order <p>For more detailed entity information you can check section 2 “Data Architecture”.</p> <p>Most of the data will be stored in the relational database. Others will be stored in</p>	<p>All user stories.</p>

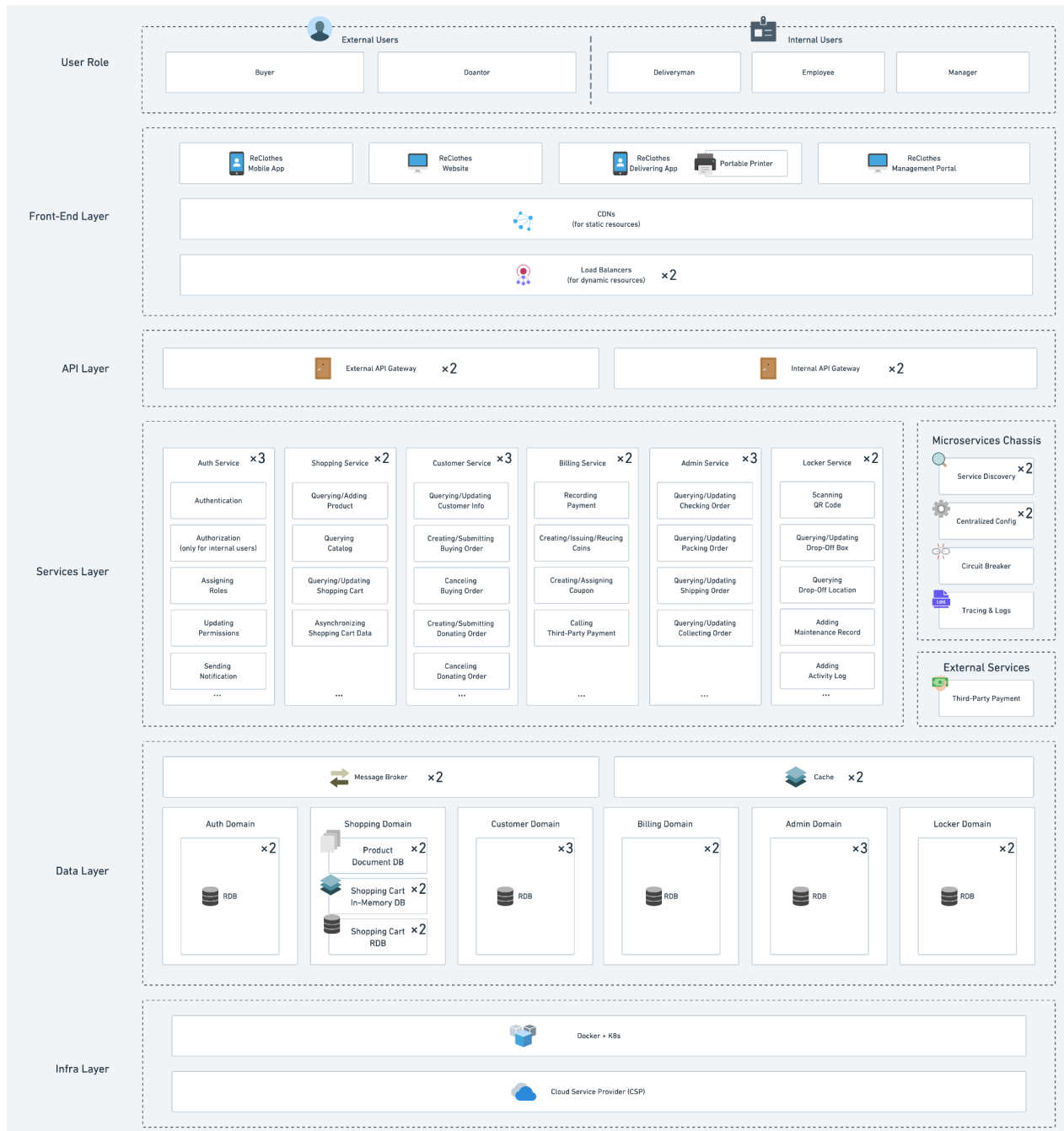
	<p>NoSQL databases. The choices of databases are based on non-functional requirements.</p> <p>For example, the shopping cart domain needs high availability, so the data of shopping carts will store in an in-memory database, and also use a message queue to transfer data asynchronously to a relational database for data persistence.</p> <p>Cause we reject any kind of data coupling between different domains, there are no foreign key constraints between different domains. Without foreign keys, we only store related ID values in the table for querying required information later.</p>	
Support Module	<p>The support module provides all kinds of services that are used by other services. Such as authentication and authorization services, notifications and logging services, message queue, and caching.</p>	
Docker & K8S	<p>The whole system will deploy in containers and be orchestrated by Kubernetes. We will use horizontal pods autoscaling to scale out or scale in pods by the traffic loads. In addition, Prometheus and Grafana will gather some key metrics from each pod to generate a dashboard for monitoring the whole system.</p>	

1.3 OUTLINE THE TYPE OF CONNECTIONS AND INTEGRATION PATTERNS

Connector Name	Description
Buying	Buyers can use ReClothes app or website to browse all kinds of used clothes, then create a buying order from their shopping cart. Later through this app or website, they can get the progress of their buying order and get some stimulation (sustainability coins or off-line coupons).
Donating	Donators can use ReClothes app or website to complete their donating form by typing detailed information about their used clothes. After that, they can use this app and website to track the progress of their used clothes and get some stimulation (sustainability coins or off-line coupons) by the condition of their used clothes.
Shipping Order Collecting Order	<p>Delivery men can use ReClothes Delivering app to track the progress of their shipping orders and collecting orders. Through this app, they can get notifications timely. In addition, this app has the ability to connect with a portable printer to print out a tracking code sticker sticking on top of the package for further tracking. By the way, delivery men must open their VPN firstly, then they can log in and use all functions of the app.</p> <p>Shipping Order: the tasks are sent to the delivery men to let them ship packages from the company to drop-off locations.</p> <p>Collecting Order: the tasks are sent to the delivery men to let them collect packages from the drop-off locations to the company.</p>
Checking Order Packing Order Product Data	<p>Employees can use ReClothes Mgt. website to manage all information through the internal network. If they want to connect to this mgt. the website from the outside network, they can use VPN tool to connect to the internal network firstly.</p> <p>Different employees with different roles will have different permissions to check or update information on the management website. Only managers have permission to assign roles or update permissions.</p> <p>Employees with permission of managing the checking order can track their assigned checking orders and update the status of their checking orders.</p> <p>Employees with permission of managing the packing order can track their assigned packing orders and update the status of their packing orders.</p> <p>Employees with permission of creating product items can create or update the information on products except for price data.</p>
Product Price Data	Only employees with manager roles can update the price of products or manage information of roles and permissions.

Roles & Permissions	
CRUD	<p>All components will finally interact with databases through the operation of creating, reading, updating, or deleting.</p> <p>Some databases will have tremendous read operations and some may have much more write operations. Based on the non-functional requirements, components will read from in-memory databases for improving the read performance, and components will use the message queue to process data asynchronously for improving the write performance.</p>
Authentication & Authorization	<p>All requests will be checked the authentication and authorization firstly after entering into the system.</p>
Notification & Logs	<p>Notifications send out to customers or employees based on the requirements.</p> <p>All business logs and metrics will record into the file system for debugging and monitoring.</p>
Caching & Asynchronization	<p>For improving the read performance, some components will read from the cache.</p> <p>For improving the write performance, some components will write to the message queue, then consumers read data for further data persistence.</p>

1.4 ARCHITECTURE STYLES USED



digital version: <https://whimsical.com/event-driven-microservice-architecture-632fvJH2g2yv5xLvT4GrHT>

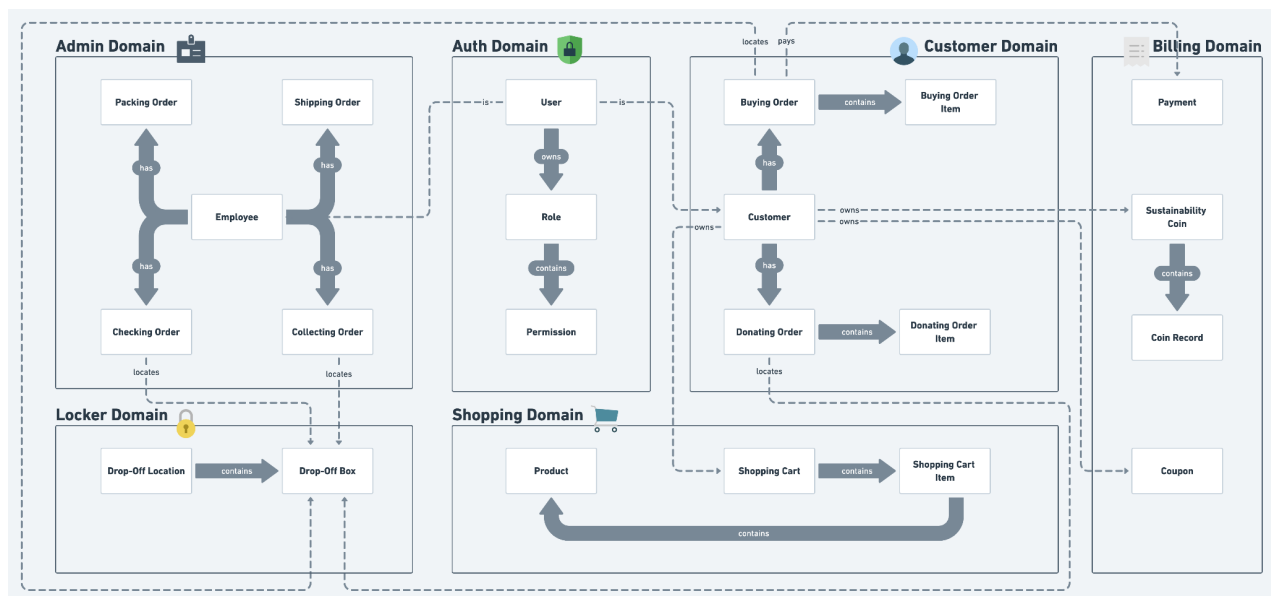
Component	Description
Front-End Layer	<p>In the front-end layer, we provide different accessing methods for different users. For example, customers can use the ReClothes mobile app or ReClothes website to buy used clothes or donate their used clothes. ReClothes delivery men can use ReClothes delivering the mobile app to track the collecting orders and the shipping orders immediately, they can also use this app to connect with their portable printer to print out tracking code stickers. Employees can use ReClothes Mgt. website to manage or update the status of their orders.</p> <p>In the front-end layer, no matter website or mobile app, they only provide functions of visualization, not provide any kind of business logic. All business logic appears in the service layer.</p> <p>In this layer, for giving customers more smooth experiences, we put some static resources into the CDN to improve the speed of loading pages.</p> <p>Next is the load balancer for processing the dynamic requests. And in this layer, requests will be forwarded to the internal API gateway if requests belong to employees, managers, or delivery men. Then other requests will be forwarded to the external API gateway if requests belong to customers.</p> <p>In the load balancer, all access logs will transfer to a log system for debugging and monitoring.</p>
API Layer	<p>API layer consists of two kinds of API gateway. One for processing internal requests, and another one for processing external requests.</p> <p>In the internal API gateway, all requests must be checked its authentication and its authorization firstly, then transfer to a specific service.</p> <p>In the external API gateway, some requests can transfer to services directly without any authentication, but some requests can transfer to services after authentication. For example, browsing products don't need any authentication check, but submitting a buying order will need to be authenticated firstly.</p> <p>By the way, for authentication and authorization, the API gateway needs to cooperate with an auth service in the service layer.</p> <p>Excepting authentication and transferring requests, API gateways can also do some requesting limitations. For example, the API gateway can reject requests after overwhelming traffic loads coming from a specific IP address.</p>
Service Layer	<p>This is the core of the whole system, all business logic is stored in this layer. After authentication and authorization, requests will be forwarded to a specific service by the API gateway.</p> <p>In this layer, for building bounded contexts, all services will separate by different domains. In its own bounded context, codes will only couple with each other,</p>

	<p>but decouple with the outside. The bounded context is just like a wall to isolate different contexts and protect data and codes that will not be influenced by other services. So, each service will acquire the ability of fault tolerance naturally. But in each service, they still need some information from the outside, each service can get data through REST API between each other. And each REST API is idempotent to call repeatedly in the condition of unstable networks.</p> <p>There also exist some basic services for supporting all other services.</p> <p>For example, discovery service is aimed at providing the ability to detect newly added services and manage the life cycle of each service.</p> <p>The Central configuration is aimed at providing a configuration center for all services, and developers can dynamically change some configurations without rebooting a specific service.</p> <p>The circuit breaker gives each service the ability to process continuously after receiving errors or nothing from another service.</p> <p>The tracing component is aimed at providing the ability to check the whole calling chain in debugging.</p> <p>The Metrics and logs give the whole system abilities to monitor the status of each virtual instance or the business logic.</p> <p>The notification component is aimed at sending out emails or App Push Service.</p>
Data Layer	<p>For pursuing high degrees of decoupling, all data belonging to different services will also be decoupled into different domains. So, the foreign key constraints only exist among the domain itself but do not exist between different domains. We only store an id value to give some weak connection with other entities outside of the domain.</p> <p>As different service has different non-function requirements, some data might be stored in a relational database for strong consistency, some data might store in an in-memory database for high availability, and some might be stored in a document database for flexibility. Others will store in a relational database if there are no special non-function requirements.</p> <p>By the way, there are no transactions that appeared between different domains cause processing distributed transactions is really expensive than designing better granularity for each service. But there still exist transactions within the domain.</p> <p>In the data layer, for resolving the bad writing performance, we use the message queue to persist data asynchronously. And for improving the read performance, we use a cache to retrieve data rapidly.</p>

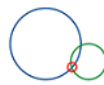
Infra Layer	<p>The whole system will deploy in a Kubernetes cluster and set up some horizontal pods scaling policies to resolve a mass of traffic load in a short time.</p> <p>Since Kubernetes has already become an industry standard, it's really easy to deploy your codes in a Kubernetes cluster in the different cloud service providers without changing any codes.</p>
--------------------	---

2 DATA ARCHITECTURE

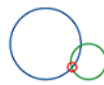
2.1 IDENTIFY DATA ENTITIES FOR EACH USER STORY



digital version: <https://whimsical.com/data-architecture-UWP6TRJeVgKASRNwZpPnsd>



User Stories	Entities	Description
epic-01: #c-01, #c-02, #c-03, #c-04, #c-05, #c-06, #c-07, #c-08, #c-09, #c-10 epic-02: #d-do-02, #d-co-04, #d-po-03, #d-po-09 epic-03: #b-bu-05, #b-pa-03, #b-po-01, #b-po-03 epic-04: #rd-c-04, #rd-r-pa-01, #rd-r-po-01, #rd-r-po-04 epic-05: #rb-r-re-03, #rb-r-co-01, #rb-r-po-01 epic-06: #mf-eu-01, #mf-eu-02, #mf-eu-03, #mf-eu-04, #mf-eu-05, #mf-eu-06, #mf-iu-01, #mf-iu-02	User, Role, Permission	User: This entity is used for authentication. The entity will store username and password for the login function of internal users and external users. Role: Each internal user owns a different role that will give them different permission for the management. In addition, only internal users own roles. Permission: The permission entity represents how many functions of this system the internal user can use. One role can have multiple permissions.
epic-02: #d-po-05 epic-05: #rb-r-po-02 epic-06: #mf-iu-01, #mf-iu-02	Employee	Employee: This entity represents internal users (employees, managers, and delivery men), and employee entity stores some data related to an employee, such as employee ID, the contract number, employment date, etc. In addition, this entity extends more functions than the user entity, and some other entities will only connect with employee entity not the user entity.
epic-06: #mf-eu-01, #mf-eu-02, #mf-eu-03, #mf-eu-04, #mf-eu-05, #mf-eu-06	Customer	Customer: This entity represents external users (buyers and donators), and it extends more functions functions than the user entity.
epic-03: #b-bu-02, #b-bu-03 epic-05: #rb-c-01, #rb-r-re-01 epic-06: #mf-bo-01, #mf-bo-02, #mf-bo-03	Buying Order, Buying Order Item	Buying Order: Storing some key information of a buying order, and connecting with the specific customer. This entity also stores the ID number of drop-off box. Buying Order Item: Product id number, the amount, product price and some



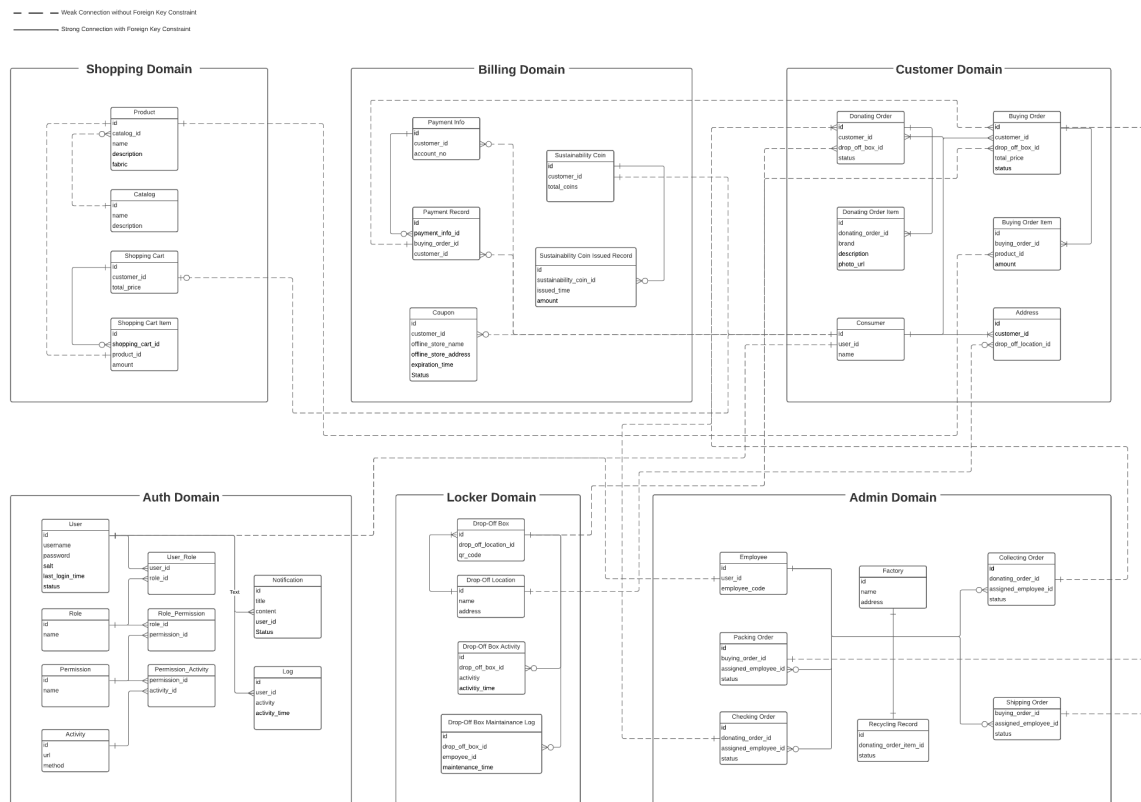
		other detailed buying order information will store in this entity.
epic02: #d-pr-01, #d-pr-02, #d-pr-04 epic-04: #rd-c-01, #rd-r-re-01 epic-06: #mf-do-01, #mf-do-02, #mf-do-03	Donating Order, Donating Order Item	Donating Order: Same with buying order, but this entity is only used for donating process. Donating Order Item: Storing some detailed donating order information.
epic-02: #d-po-01, #d-po-02 epic-06: #mf-cho-01, #mf-cho-02, #mf-cho-03	Checking Order	Checking Order: This entity is used for tracking the progress of checking the used clothes. After finished checking process, the status of this entity will be changed by the employee.
epic-03: #b-pa-01, #b-pa-02 epic-06: #mf-po-01, #mf-po-02, #mf-po-03, #mf-po-04	Packing Order	Packing Order: Through this entity, employees can track the progress of packaging.
epic-03: #b-sh-01 epic-06: #mf-so-01, #mf-so-02, #mf-so-03, #mf-so-04	Shipping Order	Shipping Order: Delivery men use this entity to acquire shipping tasks. And shipping order means that delivery men should ship packages from the company to the drop-off location.
epic-02: #d-co-01, #d-co-03 epic-06: #mf-co-01, #mf-co-02, #mf-co-03, #mf-co-04, #mf-co-05	Collecting Order	Collecting Order: A reverse version of shipping orders. It means that delivery men should collect packages from the drop-off location to the company.
epic-02: #d-po-06, #d-po-07, #d-po-08 epic-03: #b-pr-01 epic-05: #rb-r-po-04 epic-06: #mf-op-01, #mf-op-02, #mf-op-03, #mf-op-04, #mf-op-05	Product	Product: It is used for product browsing. And internal users can use this product to publish a used clothes into our online store or change the price of the used clothes.
epic-03: #b-pr-02, #b-bu-01	Shopping Cart, Shopping Cart Item	Shopping Cart: This entity is used to connect with a buyer, and also store other shopping cart data, such as total

		<p>price.</p> <p>Shopping Cart Item: Using this entity to store some detailed shopping cart item information. Such as related product ID, the date of adding, the amount of the product, and etc.</p>
<p>epic-03: #b-bu-04</p> <p>epic-04: #rd-c-02, #rd-r-re-02</p> <p>epic-05: #rb-c-02</p> <p>epic-06: #mf-dob-01, #mf-dob-02, #mf-dob-03, #mf-dob-04</p>	Payment	<p>Payment: It is used to store payment record.</p>
<p>epic-02: #d-po-04, #d-po-10</p> <p>epic-03: #b-po-04</p> <p>epic-04: #rd-r-po-03</p> <p>epic-05: #rb-c-02, #rb-r-po-03</p>	Sustainability Coin, Coin Record	<p>Sustainability Coin: For stimulation, each buyer or donator only have one sustainability coin, and each coin has a total number of coins. After finishing the process of buying and donating, buyers or donators have probability to get a limited number of coin.</p> <p>Coin Record: After getting conins and updating the total number of coins in sustainability coin, the system also store a coin receiving record for a proof.</p>
<p>epic-02: #d-po-04, #d-po-10</p> <p>epic-03: #b-po-04</p> <p>epic-04: #rd-r-po-03</p> <p>epic-05: #rb-r-po-03</p> <p>epic-06: #mf-oc-01, #mf-oc-02, #mf-oc-03</p>	Coupon	<p>Coupon: The system will also send out some off-line coupons for stimulation. So whether to get sustainability coins or off-line coupons, it is based on the condition of your used clothes or how often you buy used clothes.</p>
<p>epic02: #d-pr-03, #d-do-01, #d-do-03, #d-co-02</p> <p>epic03: #b-sh-02, #b-po-02</p> <p>epic-04: #rd-c-03, #rd-r-po-02</p>	Drop-Off Location, Drop-Off Box	<p>Drop-Off Location: Through this entity, internal users can manage all related drop-off box efficiently. Typically, a drop-off location will contain a bunch of drop-off box.</p> <p>Drop-Off Box: Each drop-off box will own</p>

epic-05: #rb-r-re-02		a exclusive QR code, users no matter internal users or external users can scan this QR code to open the door of a specific drop-off box.
epic-06: #mf-dob-01, #mf-dob-02, #mf-dob-03, #mf-dob-04		

2.2 BUILD DATA ARCHITECTURE

ReClothes ERD



digital version: <https://whimsical.com/erd-R4Q42c5PWLkum8iGfzSMuq>