

GraphQL이 가져온 에어비앤비 프론트엔드 기술의 변천사

REST서비스에서 GraphQL기반 UI설계 하기



박 호 준 Senior Software Engineer



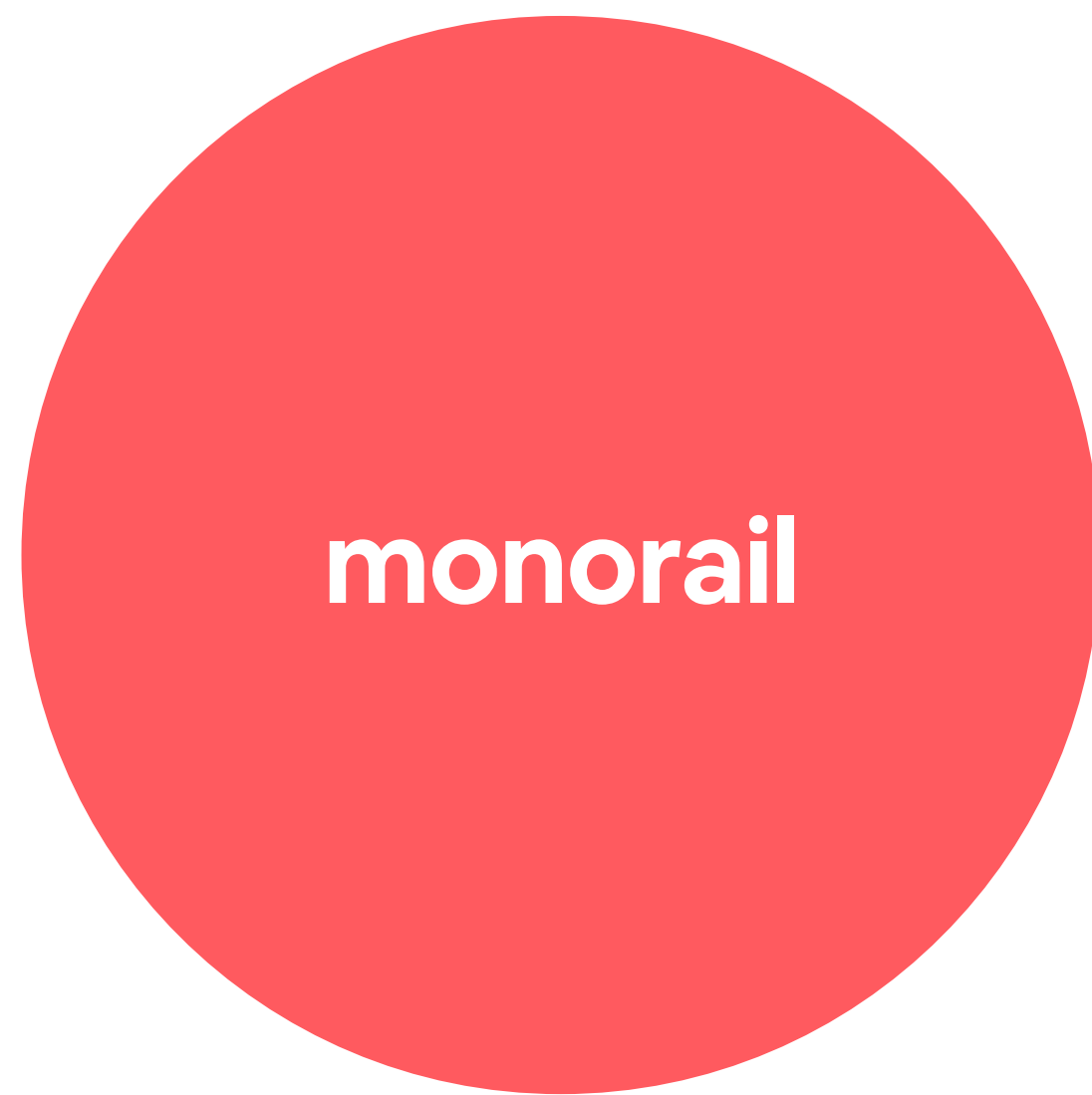
Today's Discussion

- 1.0 과거: 이전 에어비앤비 프론트엔드 시스템과 문제점
- 2.0 도입 과정: REST서비스를 유지하며 GraphQL로 이전
- 3.0 현재: GraphQL이 바뀌놓은 개발 환경
- 4.0 미래: 에어비앤비가 꿈꾸는 프론트엔드 개발환경

Airbnb Legacy System

거대한 모노리틱 아키텍처

지난 10여년간 에어비앤비는...

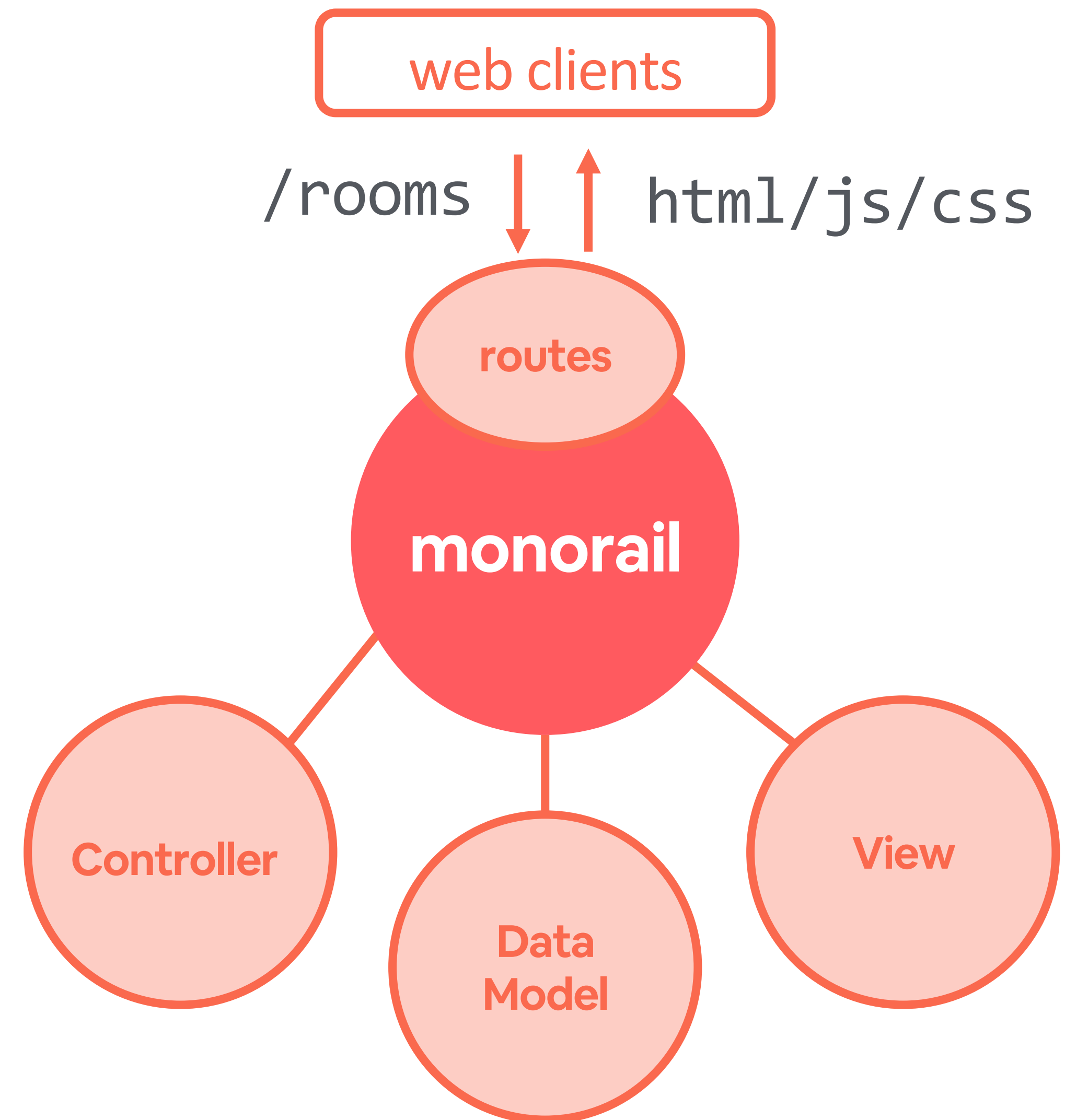


프론트엔드 구조도

전형적인 MVC구조

- Modal: ActiveModel 형태
- Controller: 라우팅 설정을 통해 관리

```
1 Rails.application.routes.draw do
2   get '/rooms' => 'listing#show'
3 end
4
5 # ...
6
7 class ListingController < ApplicationController
8   def show
9     render :index, { :bootstrap_data => {} }
10  end
11 end
12
13 # ...
14
15 <%
16   @title = "Airbnb: Legacy page title"
17   add_css_file('review_your_account')
18   render_react_component(
19     'listing.bundle.js',
20     props,
21   )
22 %>
```

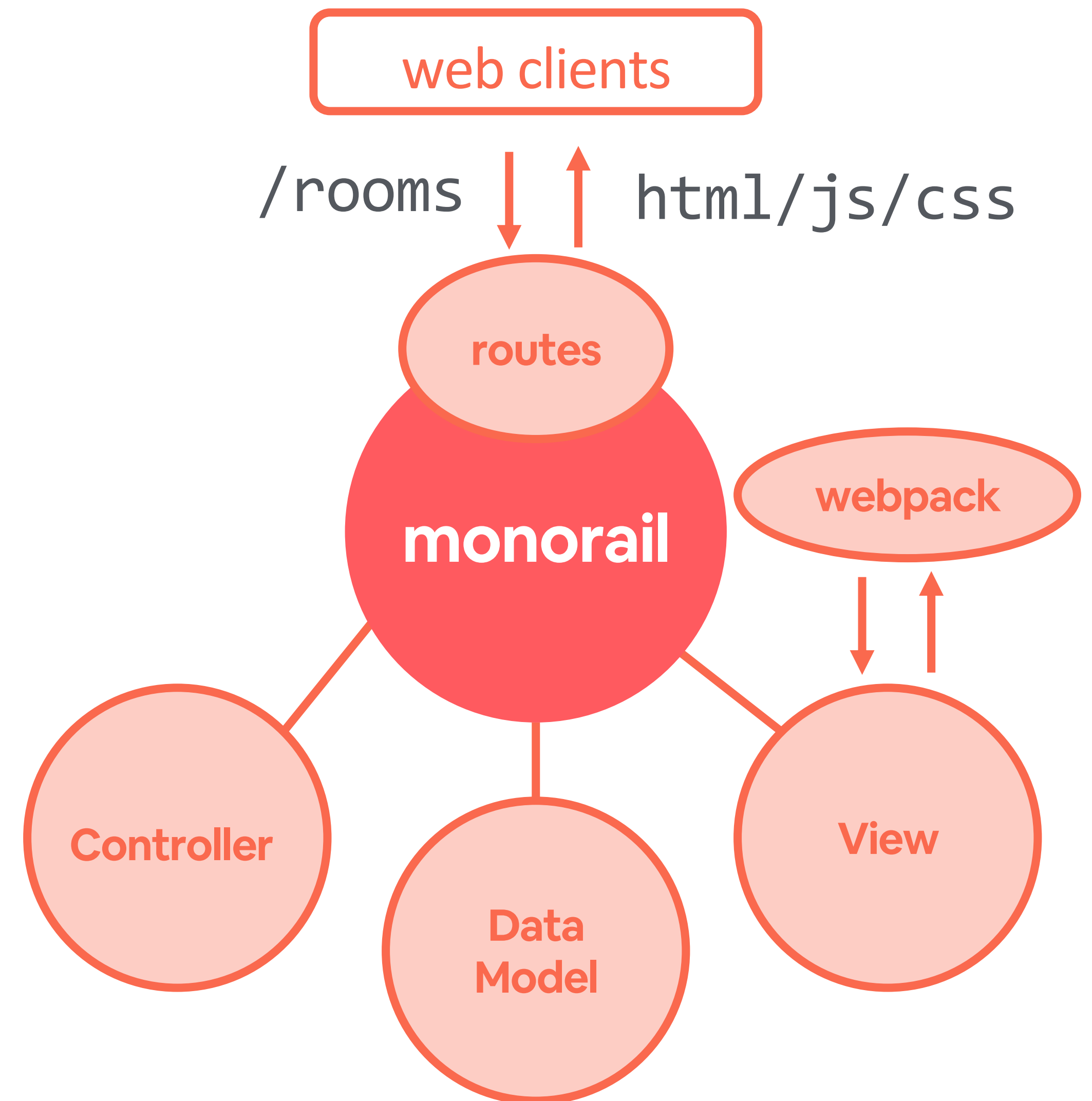


프론트엔드 구조도

전형적인 MVC구조

- View: JS번들을 사용

```
1 Rails.application.routes.draw do
2   get '/rooms' => 'listing#show'
3 end
4
5 # ...
6
7 class ListingController < ApplicationController
8   def show
9     render :index, { :bootstrap_data => {} }
10  end
11 end
12
13 # ...
14
15 <%
16   @title = "Airbnb: Legacy page title"
17   add_css_file('review_your_account')
18   render_react_component(
19     'listing.bundle.js',
20     props,
21   )
22 %>
```

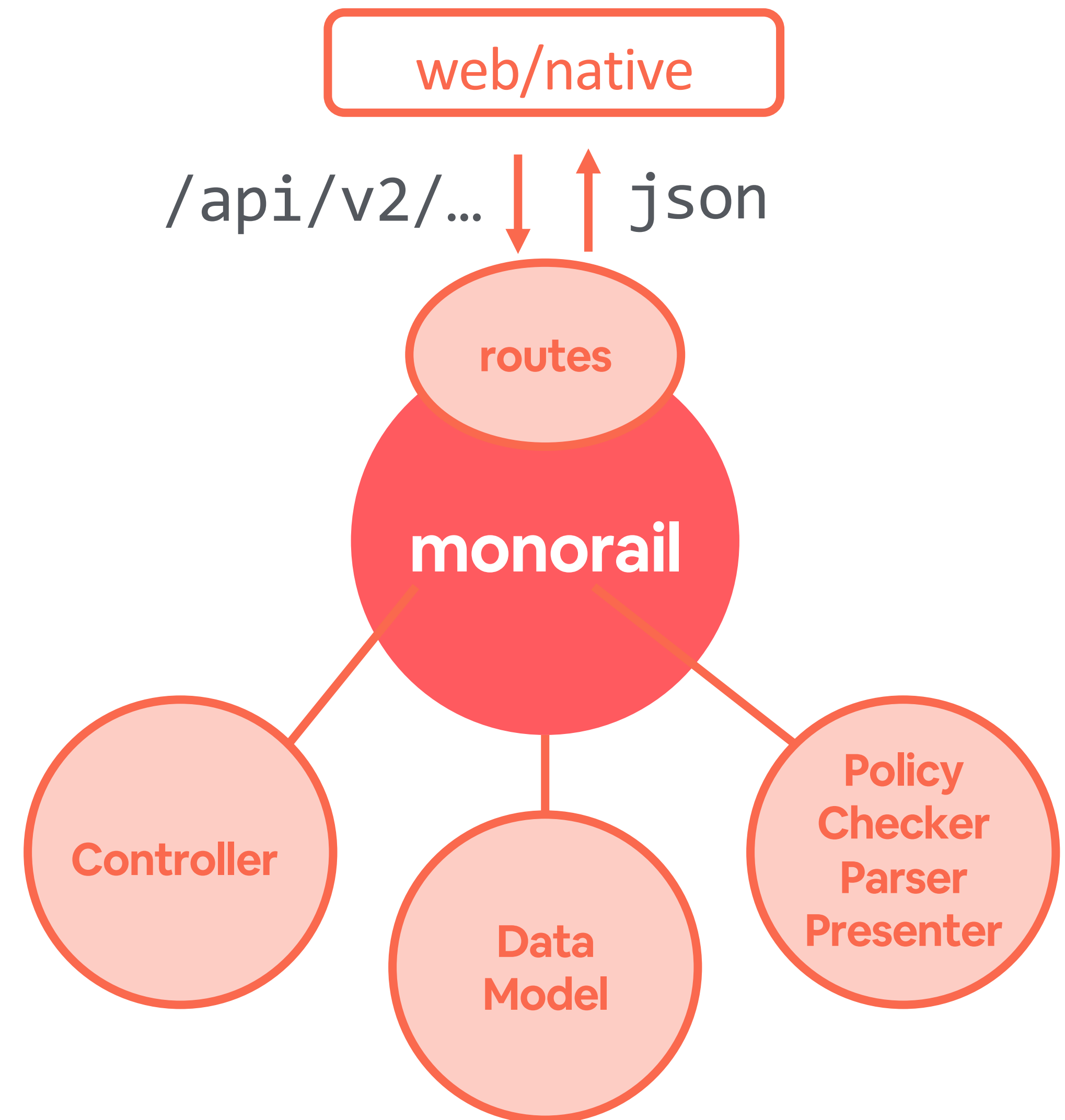


백엔드 구조도

Ruby 기반 REST 서비스

- Controller: 서비스 맵핑
- Data Model: 데이터 수집/처리
- Parser/Presenter/Policy Checker:
데이터 추출 및 권한 관리

```
1 scope 'api/v2', :module => 'api/v2' do
2   match ':resource/*id' => 'v1#show', :via => :get
3   match ':resource/*id' => 'v1#delete', :via => :delete
4   match ':resource/*id' => 'v1#patch', :via => :patch
5   match ':resource/*id' => 'v1#update', :via => :put
6   match ':resource' => 'v1#index', :via => :get
7   match ':resource' => 'v1#create', :via => :post
8 end
9
```



프론트엔드에서 API연동 과정

REST 서비스와 연동하도록 API호출 함수를 제작하여 사용

- CRUD 연동 과정을 함수로 맵핑

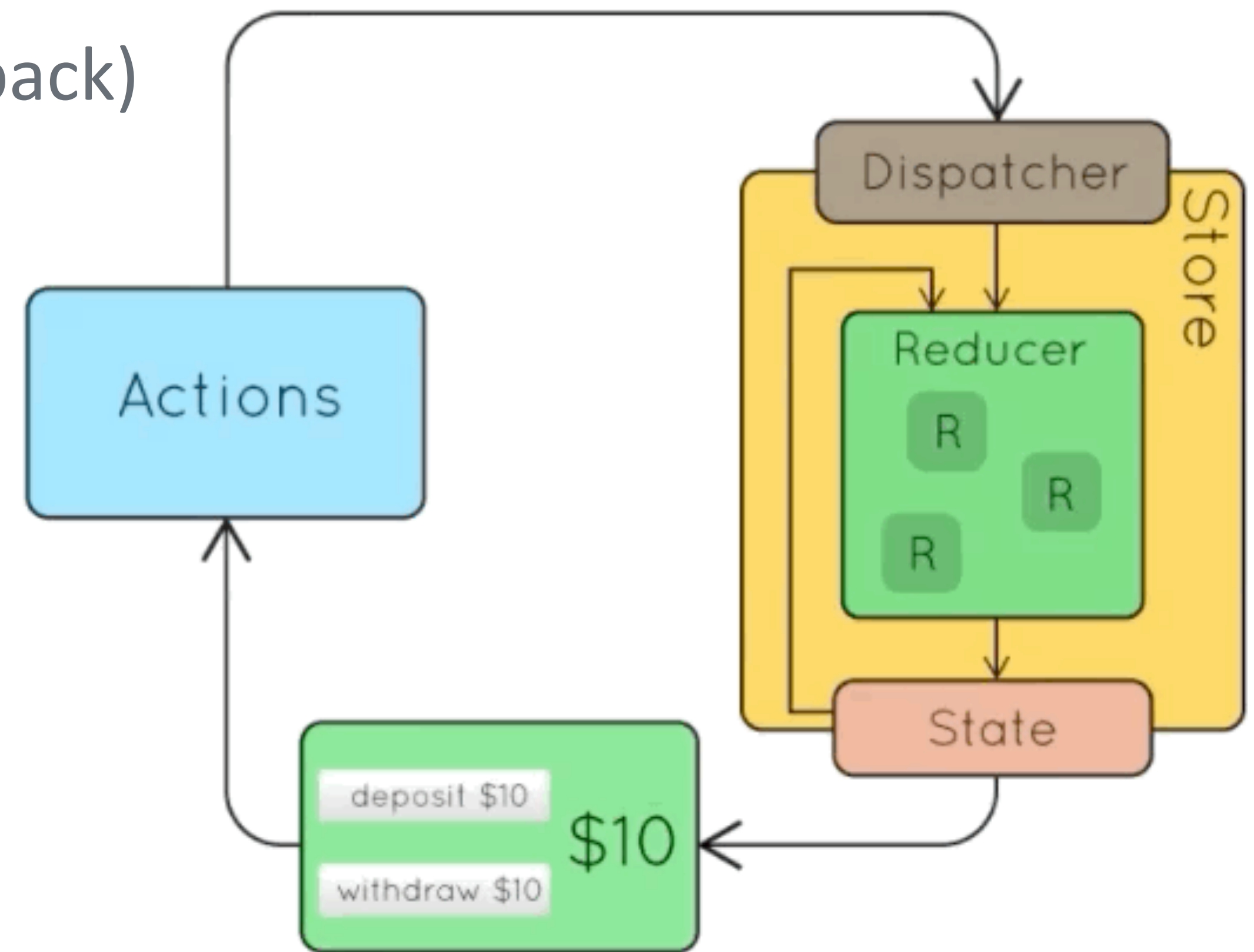
```
import { APIHost } from '@irbnb/api-resource';
const monorail = new APIHost({
  url: '/api/v2',
});
+ const userResource = monorail.resource('users');

// Calls /users with method: GET
userResource.collection();
// Calls /users/123 with method: GET
userResource.get(123);
// Calls /users with method: POST
userResource.create({}, {
  full_name: 'Jordan'
});
// Calls /users/1234 with method: DELETE
userResource.delete(1234);
```

상태 관리 라이브러리

Redux를 사용해 API요청 데이터 관리

- 미들웨어: 비동기 처리(redux-thunk, redux-pack)
- Devtools: 강력한 디버깅 기능 활용

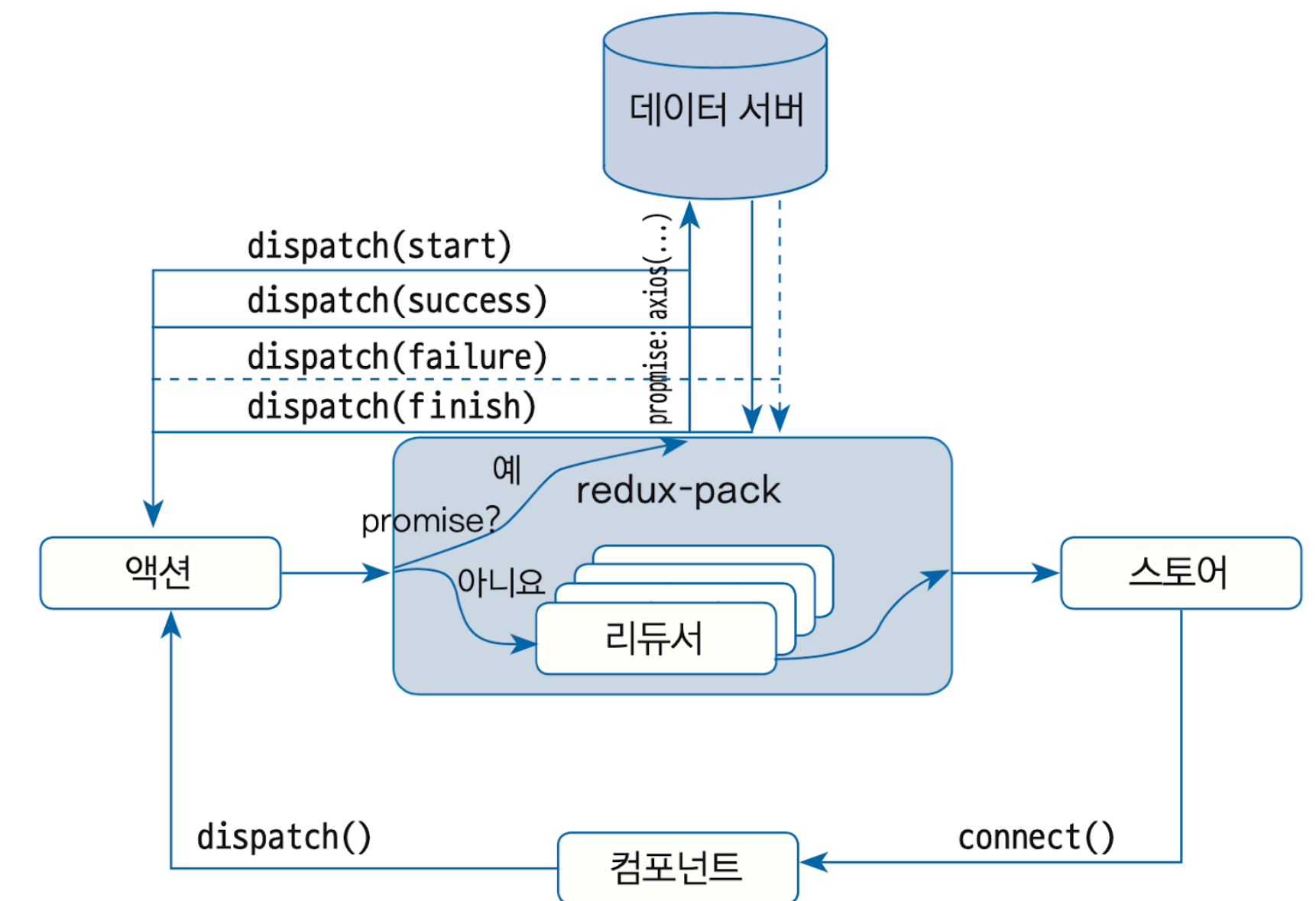


Redux와 API 연동 자동화

redux-pack 미들웨어

- 반복적인 액션/리듀서 코드를 미들웨어를 활용하여 동일하게 처리

```
function fetchResource() {  
  return (dispatch) => {  
    dispatch({ type: 'START' });  
    Api.get('/resource').then(  
      data => dispatch({ type: 'SUCCESS', payload: data }),  
      error => dispatch({ type: 'ERROR', payload: error }),  
    ).finally(  
      () => dispatch({ type: 'FINISH' })  
    );  
  }  
}
```



```
function fetchResource() {  
  return {  
    type: 'REDUX_PACK',  
    + promise: Api.get('/resource')  
  };  
}
```


2018년 에어비앤비

1000+

전체 엔지니어링 크기

>2 millions

코드의 길이

>10k

매일 배포되는
PR 개수

성장통: 유지 보수

코드 관리의 복잡성 증가

- 다수의 팀이 하나의 파일을 관리

@listing_resource.rb
(>4,000 lines of code)

```
1 # This is a listing resource. It's pretty spiffy.
2 class ListingResource < GraphQL::ActiveRecordResource
3   include ::Upshot::Agent::Tracing
4
5   TRACED_METHODS = %(
6     should_show_listing_nickname
7     photography_job_id
8     can_create_select_inbound_application
9     ready_for_select_status
10    select_listing_progress
11  ).freeze
12
13  include Bookings::HostingExperimentExtensions
14  include ListingVerificationFrameworkHelper
15
16  SEARCH_PICTURE_URLS_LIMIT = 30
17
18  Luxury::ListingResourceCollectiveLoad.register(self)
19
20  PositiveNegativeRatingCounts = Struct.new(:positive, :negative)
21
22  internal_methods(
23    :accessible?,
24    :active?,
25    :add_to_user_result_set,
26    :amenity_ids,
27    :batch_moderate_with_keys,
28    :cache_select_opt_out_callback_action,
29    :can_manage?,
30    :can_read?,
31  )
32
33  # dependent fields are the fields stored in another table with hosting_id field to
34  # relate back to hostings table, so the listing has to be created first before we can
35  # change dependent fields
36  DEPENDENT_FIELDS = [
37    "description_properties_to_update", # description must be saved before requested_approval_status
38    "amenities_ids",
39    "amenity_categories",
40    "amenity_statuses",
41    "listing_weekend_price",
42  ].freeze
43
44  # Just like the ones above but are allowed to be set to nil
45  # All dependent fields should ideally be in this array and
46  # nilable/not nilable should be enforced in the parser
47  DEPENDENT_NULLABLE_FIELDS = [
48    "bathroom_shared_with_category",
49    "bathroom_type",
50  ].freeze
51
52  GUEST_CONTROL_FIELDS = [
53    "are_children_allowed",
54    "are_infants_allowed",
55    "children_not_allowed_details",
56  ].freeze
57
58  TRUTHY_VALUES = [
59    '1',
60    'true',
61  ].freeze
62
63  FALSY_VALUES = [
64    '0',
65    'false',
66  ].freeze
67
68  LISTINGS_TYPE_RECENTLY_VIEWED = "recently_viewed_listings".freeze
69  LISTINGS_TYPE_TOP_LISTINGS = "top_listings".freeze
70
71  CURRENCY_FIELDS = [
72    :listing_currency,
73  ].freeze
74
75  CHECK_IN_TIMES_KEYS = [
76    :check_in_time_start,
77    :check_in_time_end,
78  ].freeze
79
80  FORMATS_WITH_MASTER_READ = [
81    :for_pricing_settings_after_update,
82  ].to_set
83
84  DEFAULT_SORT_ORDER = 'has_availability DESC, updated_at DESC'.freeze
85  NAME_SORT_ORDER = 'has_availability DESC, name ASC'.freeze
86  ID_SORT_ORDER = 'id ASC'.freeze
87  REVIEW_SCORES_SORT_ORDER = 'by_review_scores'.freeze
88  LISTING_STATE_ORDER = 'by_listing_state'.freeze
89
90  DB_SORTABLE_ORDERS = [
91    DEFAULT_SORT_ORDER,
92    ID_SORT_ORDER,
93    NAME_SORT_ORDER,
94  ].freeze
95
96  UNLISTED = 'unlisted'.freeze
97  LISTED = 'listed'.freeze
98  IN_PROGRESS = 'in progress'.freeze
99  PENDING = 'pending'.freeze
100
101  ALL_STATUSES = [
102    UNLISTED,
103    LISTED,
104    IN_PROGRESS,
105    PENDING,
106  ].freeze
107
108  SORTABLE_ORDERS = DB_SORTABLE_ORDERS + [
109    REVIEW_SCORES_SORT_ORDER,
110  ]
```

← List Your Space edit

← Logging update

← Host Calendar edit

← GDPR update

← Search team edit

← Booking team edit

← Pricing Models edit

← Search Ranking edit

성장통: 통합 관리

Redux 코드 관리의 복잡성 증가

- 각각의 페이지에서 독립적인 데이터 스토어를 관리하게 됨
- 팀이 늘어나면서 스토어도 급속도로 증가하게 됨

airbnb / configureStore.js

You've activated the file finder. Start typing to filter the file list. Use **↑** and **↓** to navigate, **enter** to view files, **esc** to exit. ✕

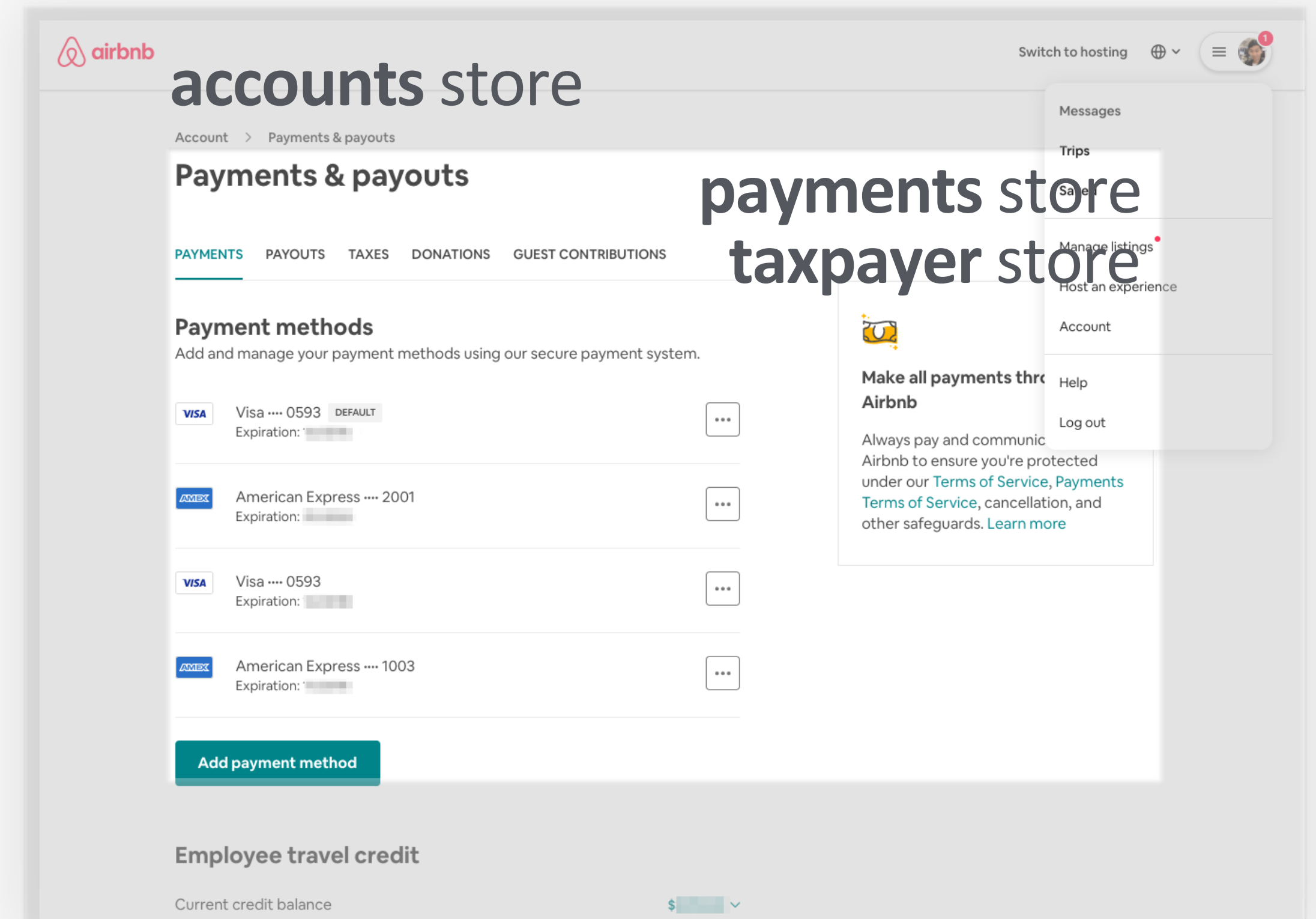
> app/assets/javascripts/fixit/store/configureStore.js
app/assets/javascripts/help_center/configureStore.js
app/assets/javascripts/homes/store/configureStore.js
app/assets/javascripts/settle_up/store/configureStore.js
app/assets/javascripts/apps/trip_issues/configureStore.js
app/assets/javascripts/instant_book/store/configureStore.js
app/assets/javascripts/apps/check_in_guide/configureStore.js
app/assets/javascripts/apps/memories/store/configureStore.js
app/assets/javascripts/apps/mt_admin/store/configureStore.js
app/assets/javascripts/apps/plan_your_trip/configureStore.js
app/assets/javascripts/account_delete/store/configureStore.js
app/assets/javascripts/listi
app/assets/javascripts/par
app/assets/javascripts/ghc
app/assets/javascripts/hel
app/assets/javascripts/eve
app/assets/javascripts/sele

```
1 import { reducer as formReducer } from 'redux-form';
2 import activeStepReducer from './activeStep';
3 import applicationReducer from './application';
4 import errorReducer from './error';
5 import paymentOptionsReducer from './payoutOptions';
6 import reservationReducer from './reservation';
7 import settleUpFormReducer from './settleUpForm';
8 import userReducer from './user';
9
10 export default {
11   activeStep: activeStepReducer,
12   application: applicationReducer,
13   error: errorReducer,
14   form: formReducer,
15   payoutOptions: paymentOptionsReducer,
16   reservation: reservationReducer,
17   settleUpForm: settleUpFormReducer,
18   user: userReducer,
19 };
```

성장통: 통합 관리

Redux 조합의 문제점

- SPA로 통합하면서 각 팀별로 분리된 스토어를 조합하는 문제점 발생

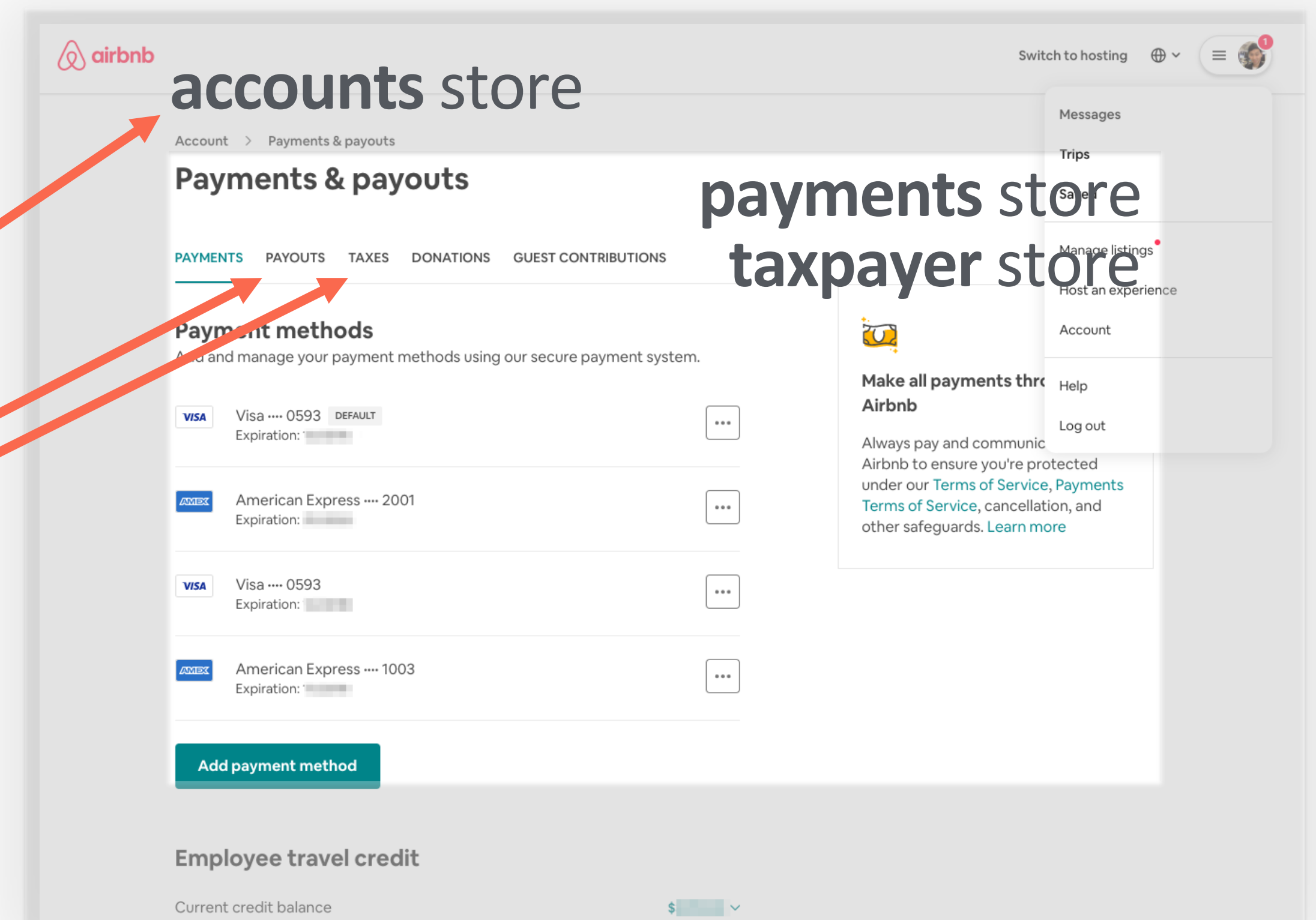


성장통: 통합 관리

Redux 조합의 문제점

- 키 충돌 방지 함수 등을 제작하여 임시 방편으로 해결

```
1 // store.js
2
3 import { createInjectableStore } from 'redux-injectable-store';
4 import { applyMiddleware, compose } from 'redux';
5 import thunk from 'redux-thunk';
6 import pack from 'redux-pack';
7
8 const initialState = getInitialState();
9
10 const enhancer = applyMiddleware(
11   thunk,
12   pack.middleware
13 );
14
15 const store = createInjectableStore({
16   initialState,
17   enhancer,
18 });
19
20 export default store;
21
22 // withReducers.js
23
24 import AccountsReducer from '../path/to/AccountsReducer';
25 import PaymentsReducer from '../path/to/PaymentsReducer';
26 import TaxReducer from '../path/to/TaxReducer';
27
28 store.injectAll({
29   accounts: AccountsReducer,
30   payments: PaymentsReducer,
31   taxpayer: TaxReducer,
32 });
```



더이상 감당하기 힘든 상황

장애 요인으로 인한 개발 지연

- Rollback 수시로 발생

Deploy Changeset							
See these changes on Github							
[redacted] please deploy to production!							
Snapshot	SHA	Author	Summary	Approvals			Notes
✓ #171859002	2f6ba3c	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Revert Deploy
✓ #171850542	27c5a1d	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171850367	233451e	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171847327	7fd4e5b	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171847224	85a861a	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171844785	1956e84	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171843771	8f48348	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171835634	224986c	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171835291	8ac4079	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171835043	fb1d0c2	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
⚠ #171834963	6165dba	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes
✓ #171832242	ac30431	[redacted]	[EMERGENCY MERG...	Revert	Checking	Good	Notes

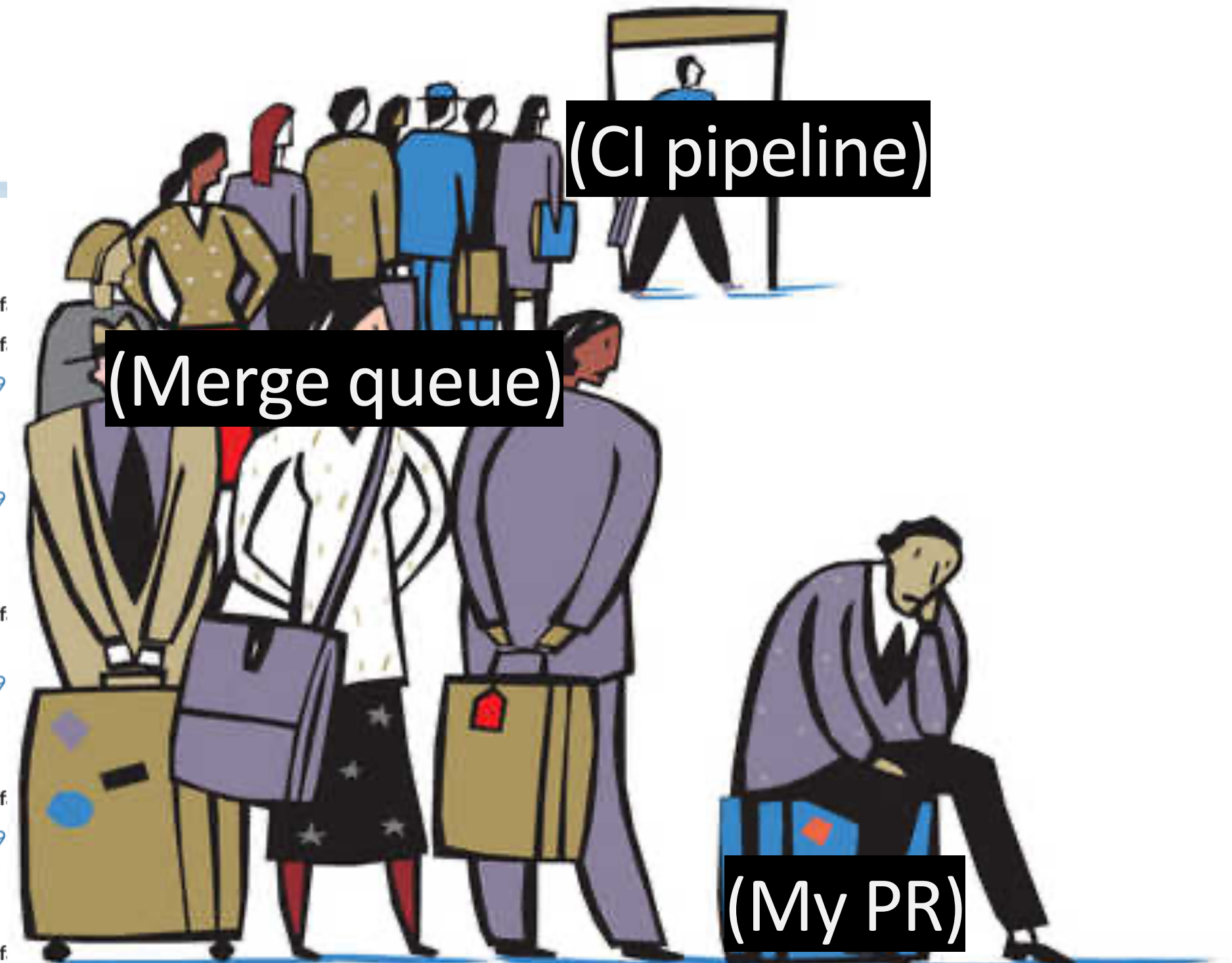
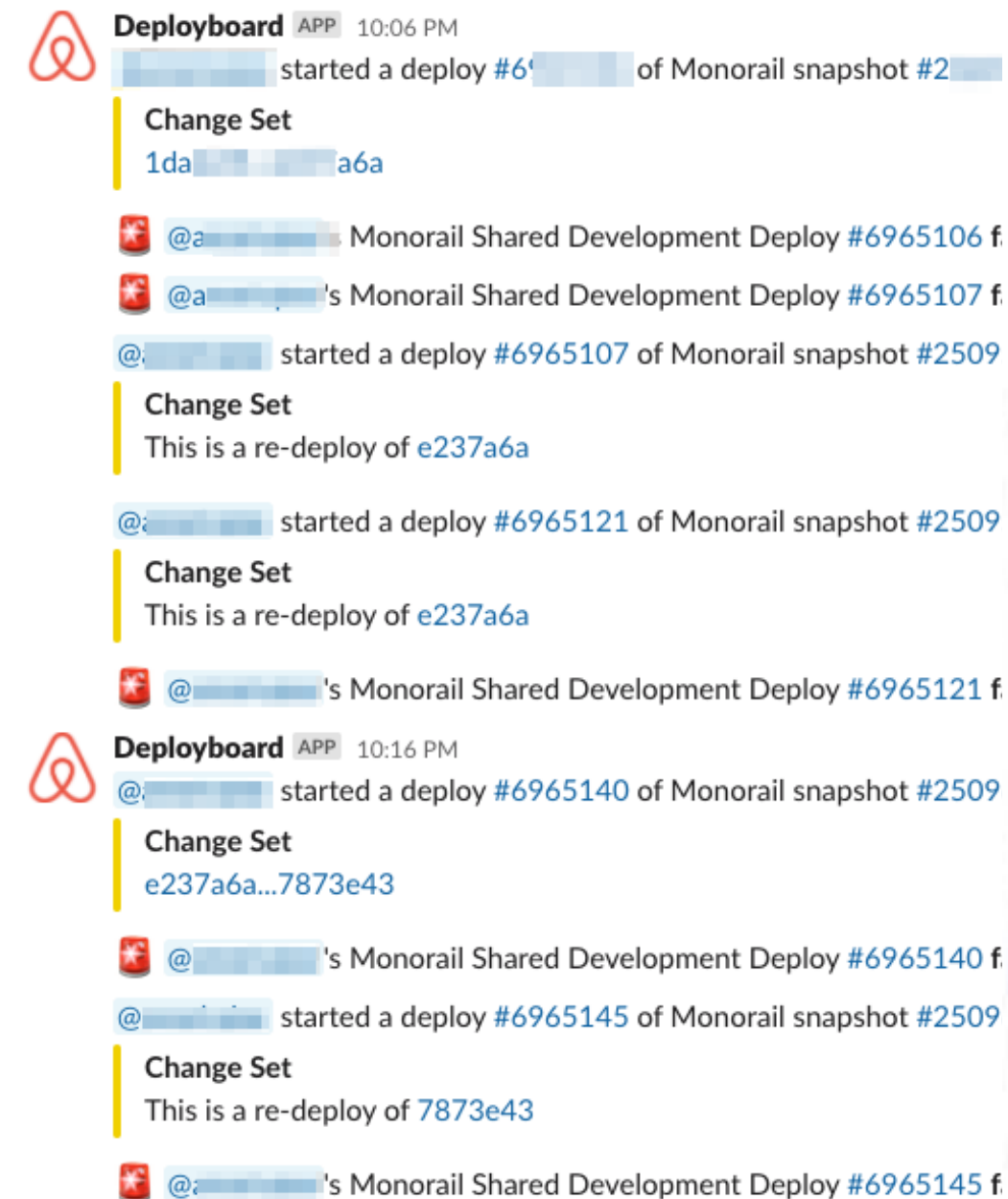
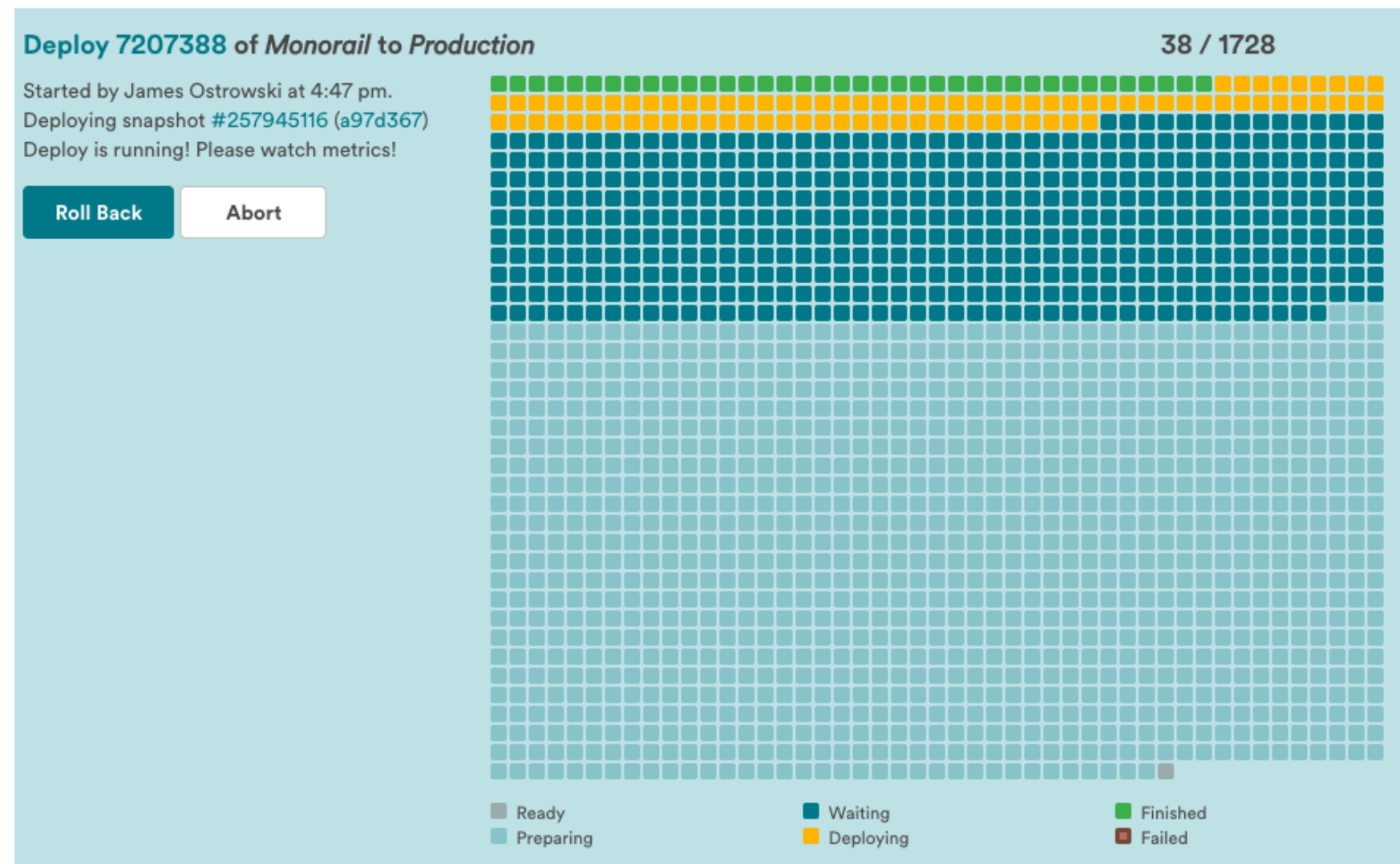
더이상 감당하기 힘든 상황

장애 요인으로 인한 개발 지연

- Rollback 수시로 발생
- Merge Lockdown, 배포 지연
- 빈번한 사건 사고

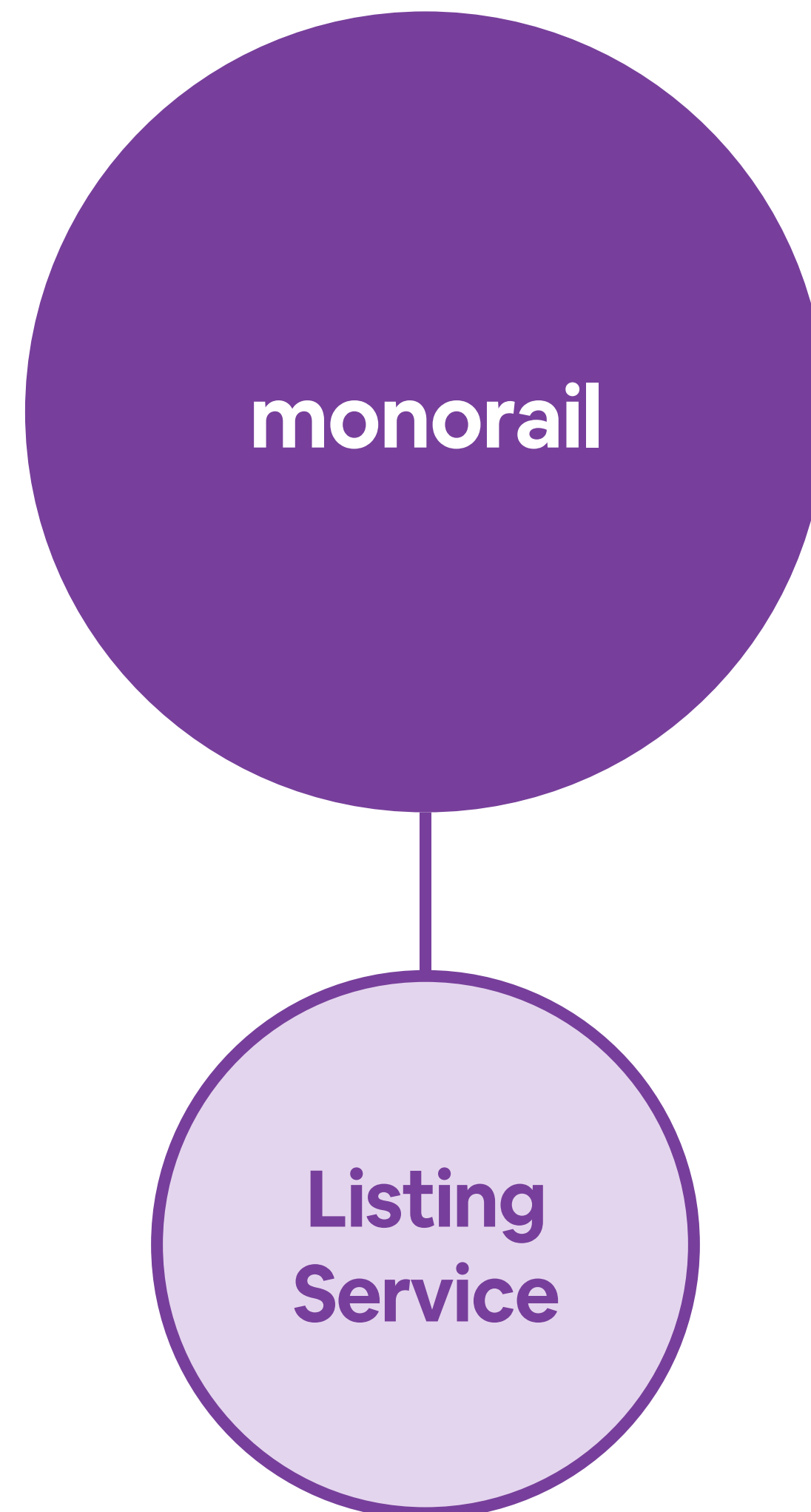
15h

매주 배포 서버가
중간되는 평균 시간

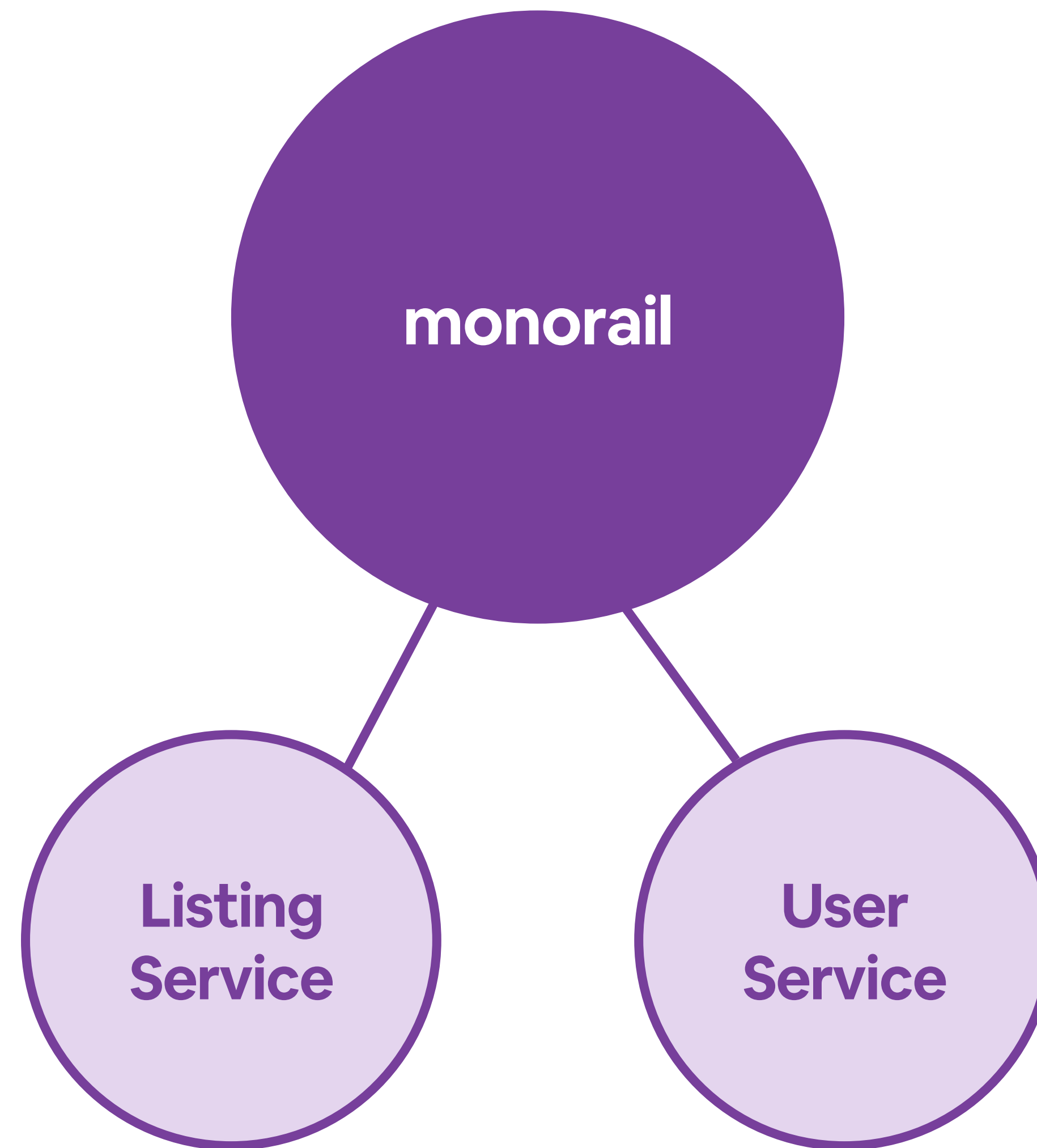


Migrate to Airbnb 2.0

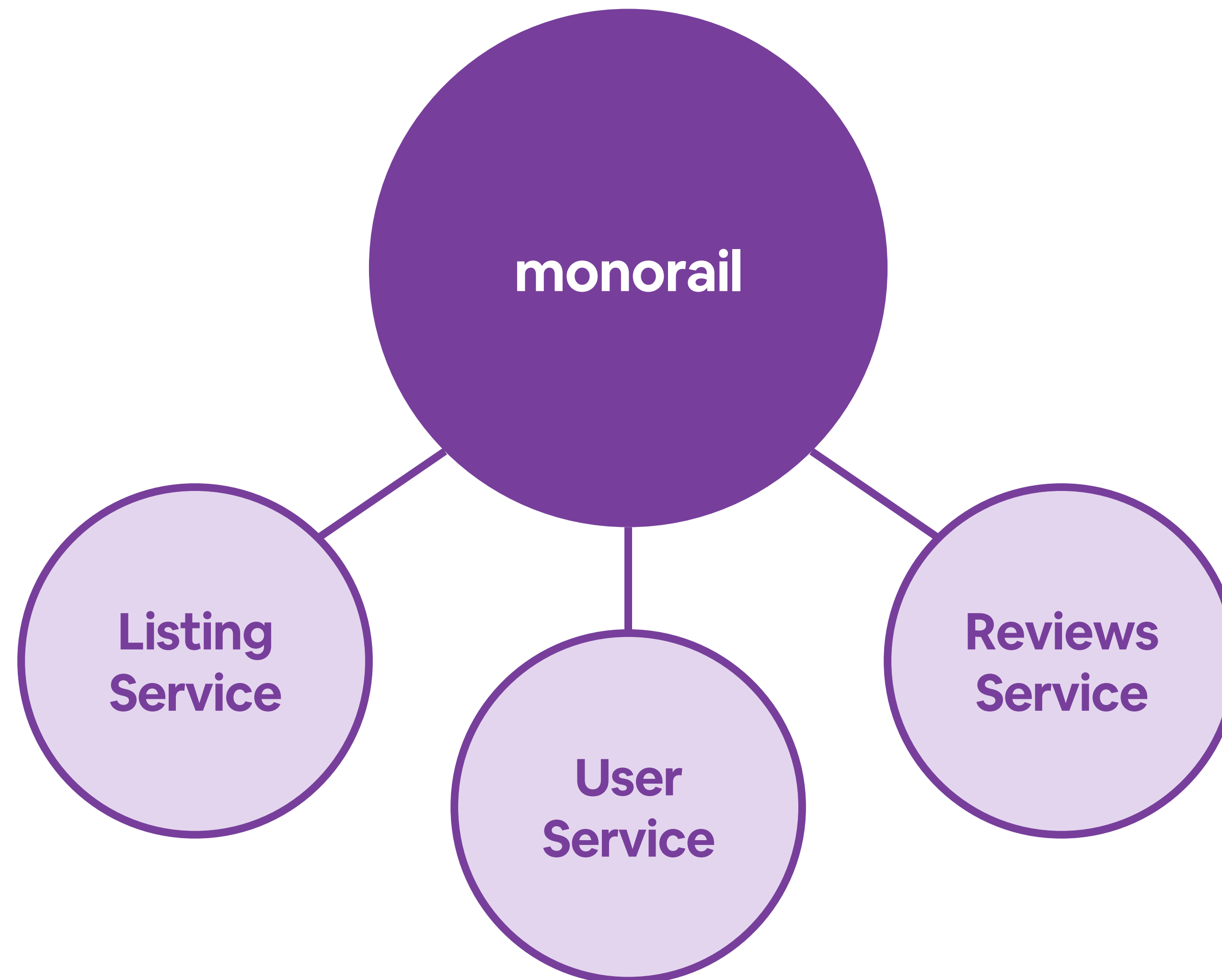
서비스 기반 아키텍처(SOA)로



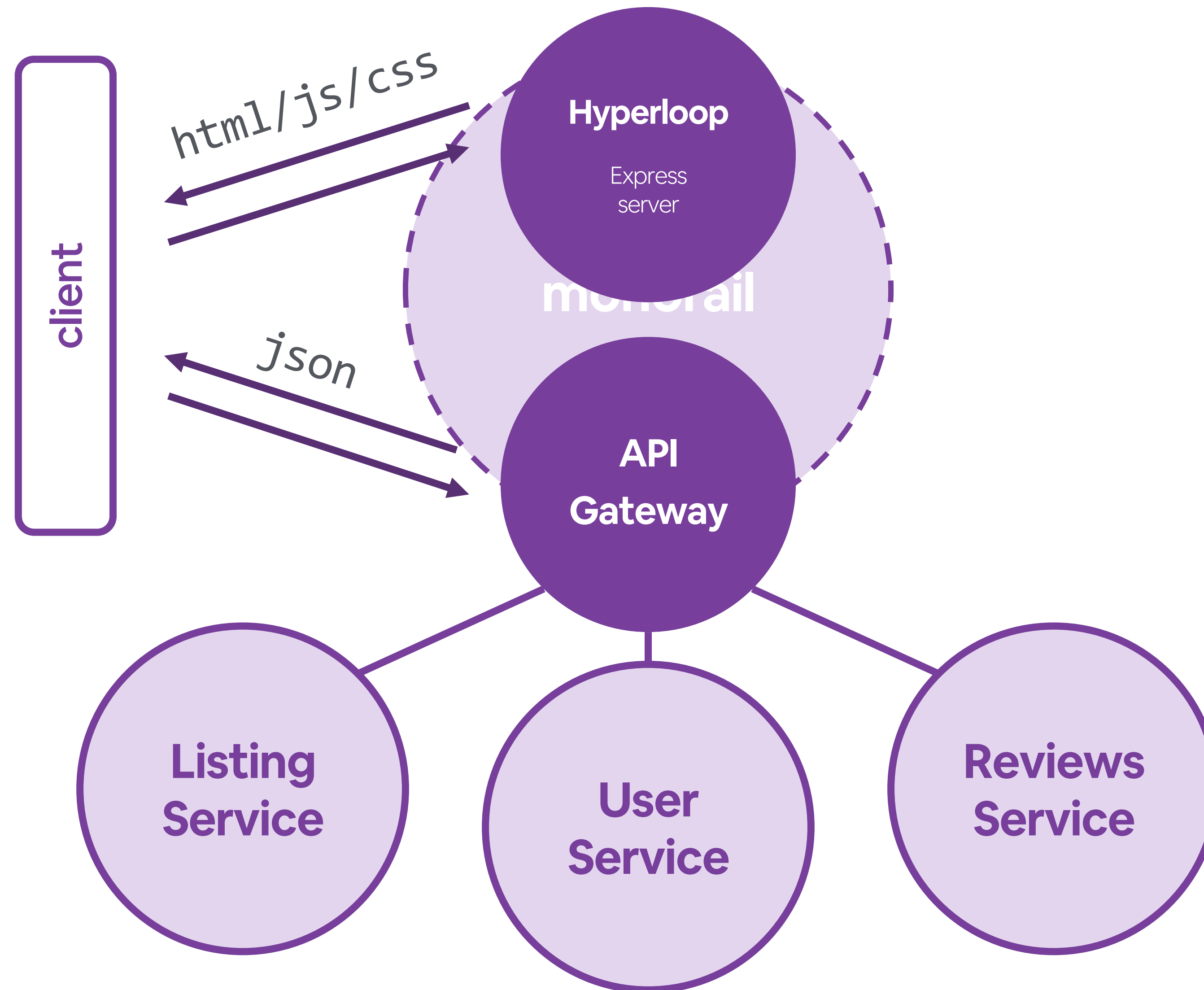
서비스 기반 아키텍처(SOA)로



서비스 기반 아키텍처(SOA)로



서비스 기반 아키텍처(SOA)로



Airbnb ♥ Thrift

Thrift

Strongly typed data object

- 타입을 필수로 선언하는 데이터 오브젝트 형태

```
1 enum Icon {
2     CHECK = 0,
3     EXCLAMATION_CIRCLE = 1,
4     BAN = 2,
5     SMAILE = 3,
6 }
7
8 struct Banner {
9     1: optional Icon icon
10    2: optional string title
11    3: optional string subtitle
12    4: optional string ctaCopy
13    5: optional string ctaUrl
14 }
15
16 struct Title {
17     1: optional string title
18     2: optional string subtitle
19 }
20
21 struct ProductCard {
22     1: optional string title
23     2: optional string subtitle
24     3: optional string imageUrl
25 }
```

Thrift IDL (Interface Definition Language)

Thrift IDL이란?

- 서비스 기능에 대한 인터페이스와 데이터 타입을 정의
- 자동 생성된 클라이언트 코드
- 각기 다른 언어의 서비스끼리 RPC를 통해 통신 가능

```
1 struct PdpRequest {
2   1: required i32 productId
3 }
4
5 struct PdpResponse {
6   1: optional i32 id
7   2: optional string name
8   3: optional i32 price
9 }
10
11 service Market {
12   PdpResponse getPdp(1: PdpRequest request)
13 }
14
```

Presentation Service Framework

Presentation service framework의 용도

- 권한 검사
- 인프라 설정 (replay, idempotent request)
- 모니터링
- REST서비스 맵핑

```
1 struct PdpRequest {
2   1: required i32 productId
3 }
4
5 struct PdpResponse {
6   1: optional i32 id
7   2: optional string name
8   3: optional i32 price
9 }
10
11 service Market {
12   PdpResponse getPdp(1: PdpRequest request)
13 }
14
```

```
1 import * as express from 'express';
2 import { createThriftServer } from '@creditkarma/thrift-server-express';
3 import { Market, IPdpRequest, IPdpResponseArgs } from './codegen';
4 //IPdpResponseArgs
5 const PORT = 8080;
6
7 const serviceHandlers: Market.IHandler<express.Request> = {
8   getPdp(request: IPdpRequest): IPdpResponseArgs {
9     return {
10       id: request.productId,
11       name: '에어컨',
12       price: 350000,
13     };
14   },
15 };
16
17 const app: express.Application = createThriftServer({
18   path: '/thrift',
19   thriftOptions: {
20     serviceName: 'market-service',
21     handler: new Market.Processor(serviceHandlers),
22   },
23 });
24
25 app.listen(PORT, () => {
26   console.log(`Express server listening on port: ${PORT}`);
27 });
28
```


Thrift IDL 활용 예제 – REST

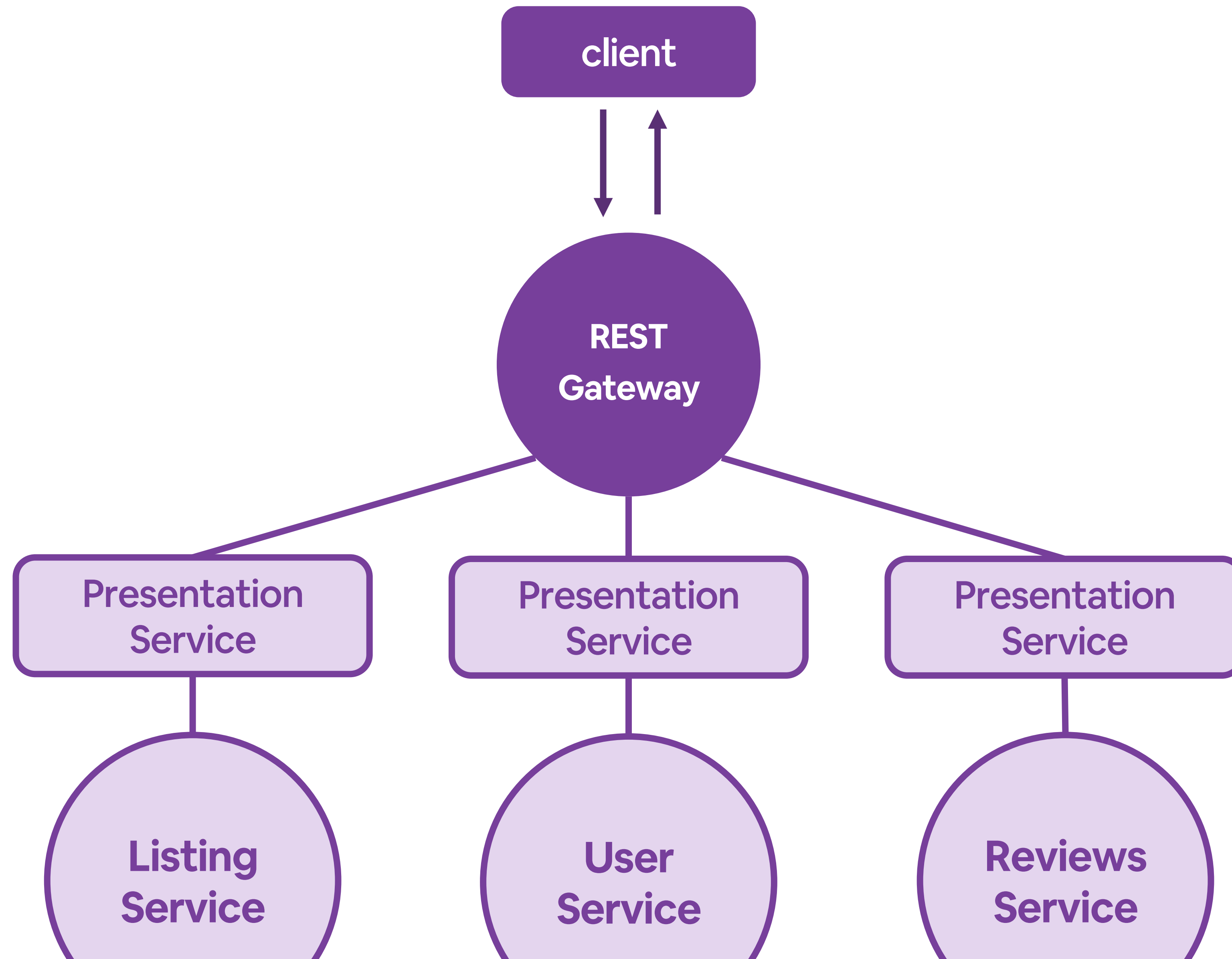
Thrift IDL과 REST서비스와의 연동 방법

- Annotation을 활용하여 자동화
- 특정 값(rest_verb, rest_path)을 활용하여 서비스 매핑을 자동화

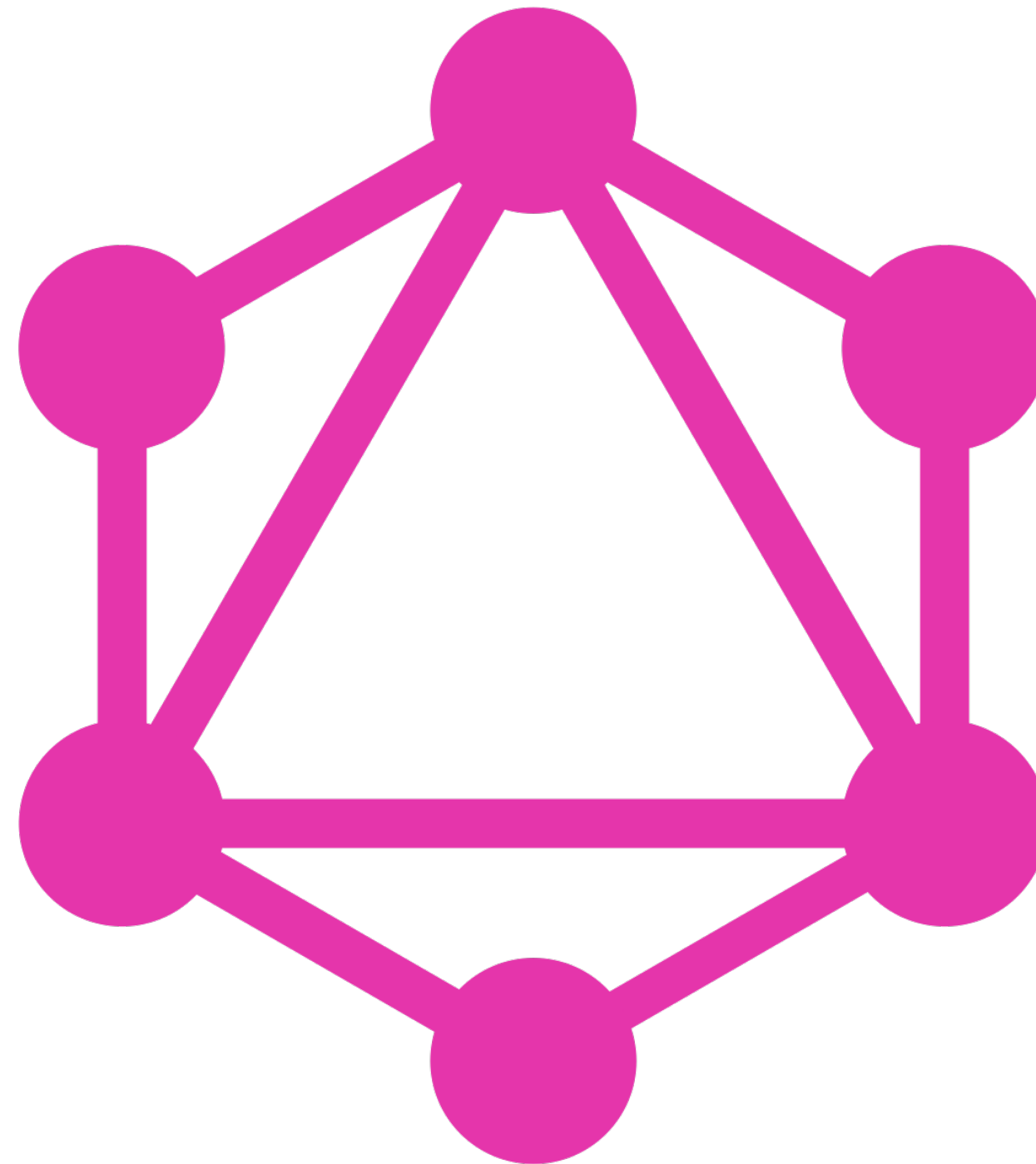
```
1 struct PdpRequest {
2   1: required i32 productId
3 }
4
5 struct PdpResponse {
6   1: optional i32 id
7   2: optional string name
8   3: optional i32 price
9 }
10
11 service Market {
12   PdpResponse getPdp(1: PdpRequest request)
13   (
14     rest_verb = "get"
15     rest_path = "/pdp/{productId}"
16   )
17 }
18
```

```
1 module.exports = function createRestServer(app, ServiceHandler, thriftClient) {
2   Object.entries(ServiceHandler.methodAnnotations)
3     .forEach(([rpcMethod, { annotations }]) => {
4     if (annotations.rest_path && thriftClient[rpcMethod]) {
5       const { expressPath, pathParts } = parsePSFPath(annotations.rest_path);
6       app[annotations.rest_verb](expressPath, (req, res) => {
7         const args = {};
8         pathParts.forEach((attrKey) => {
9           args[attrKey] = req.params[attrKey];
10         });
11         thriftClient[rpcMethod](args).then((result) => {
12           res.send(result);
13         });
14       });
15     }
16   });
17 };
```

서비스 구조도



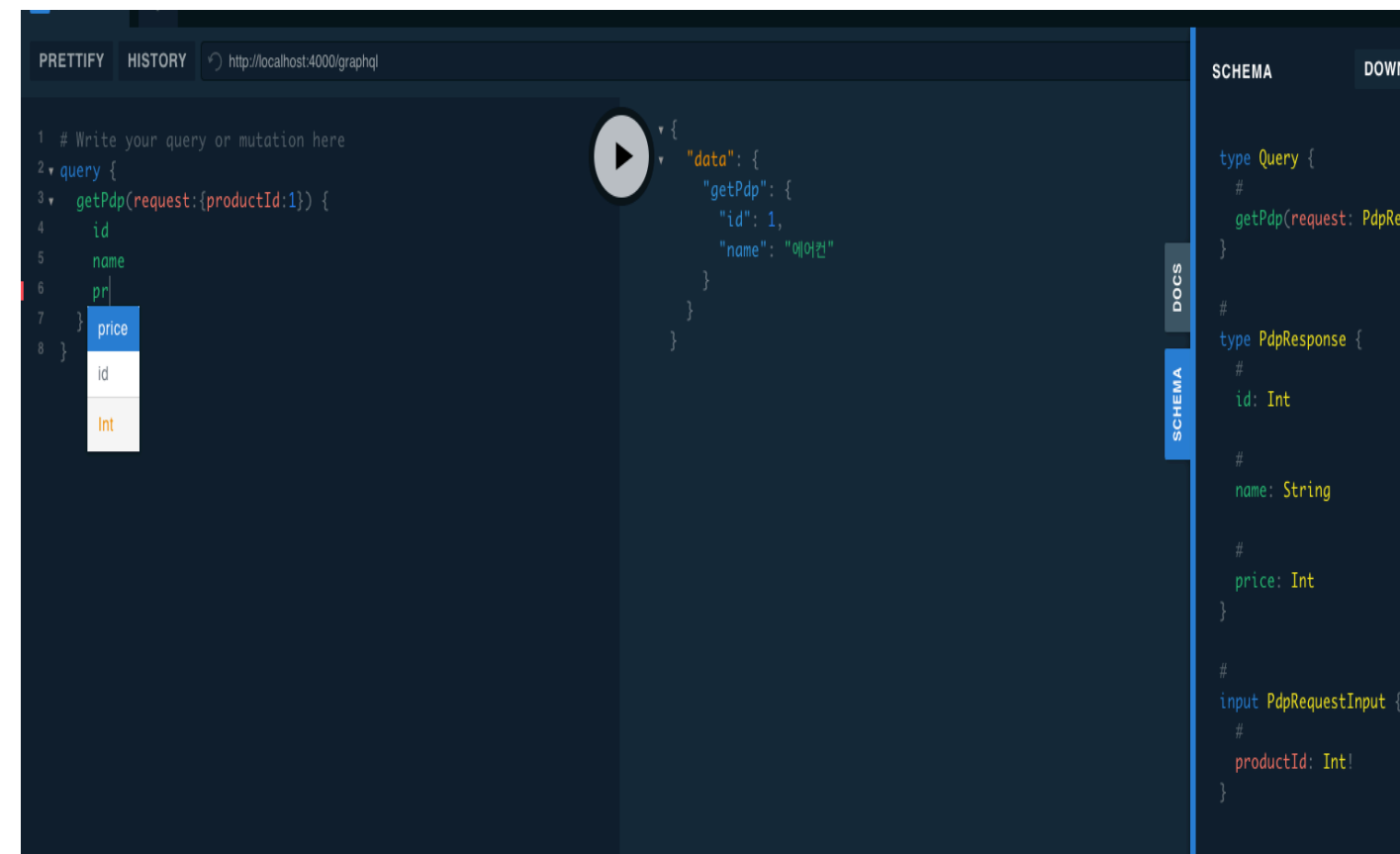
GraphQL



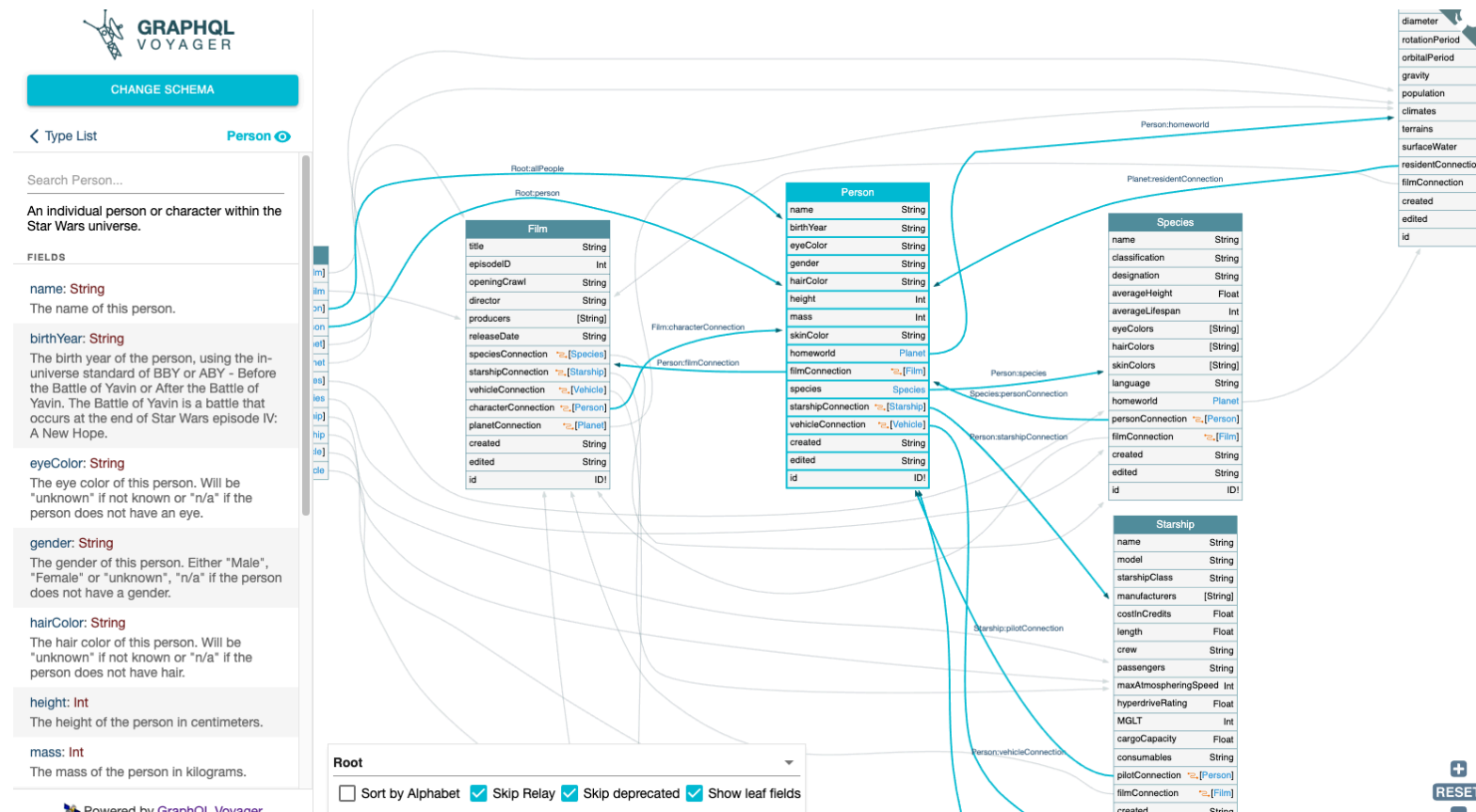
Why ?

GraphQL을 선택한 이유

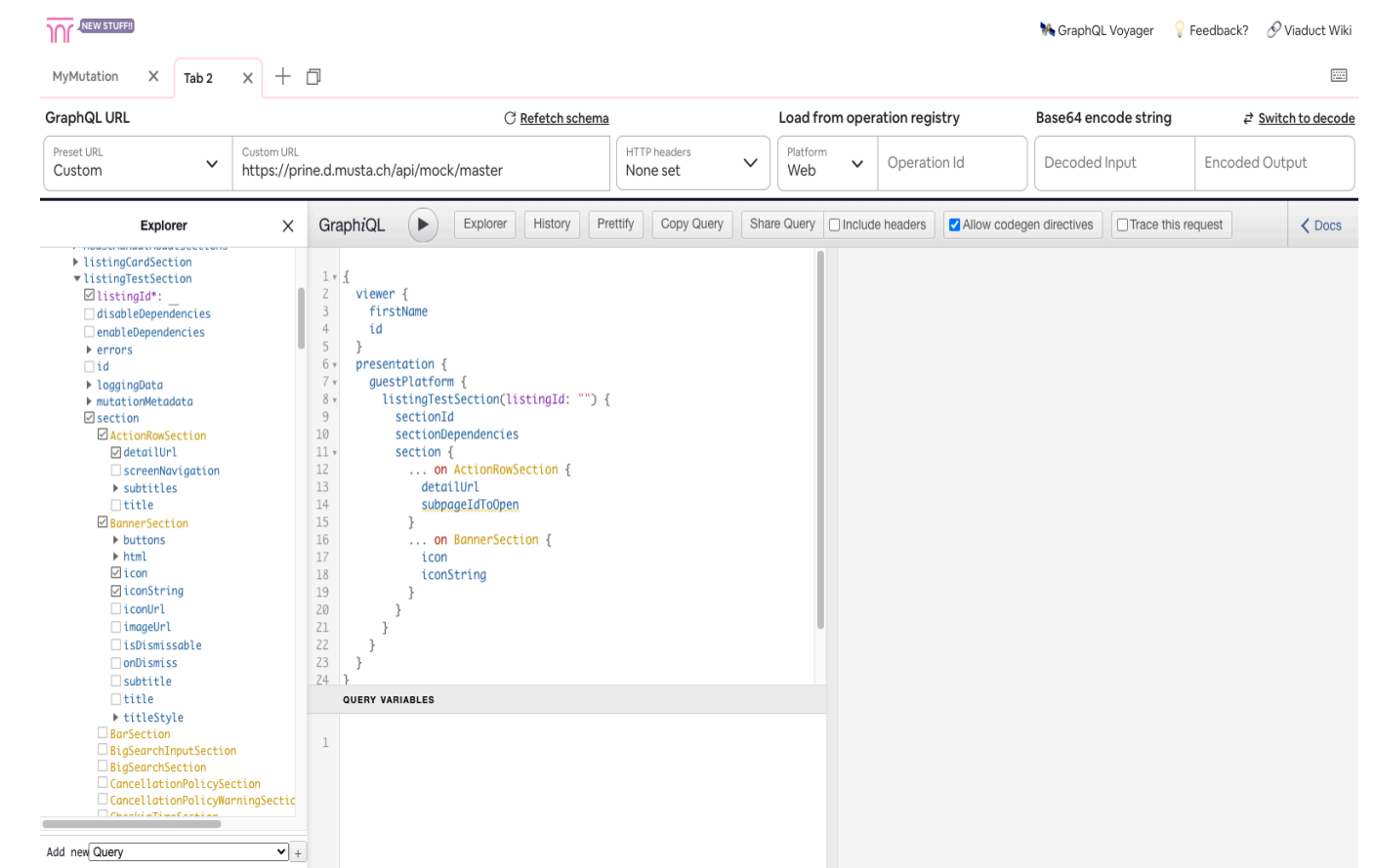
- 백앤드 <> 프론트 엔드 사이의 강력한 타입 정의 제공
- 원소스 멀티 플랫폼 (웹, 모바일, 네이티브, 툴등)
- 다양한 개발자 툴과 협업



Graphiql



GraphQL Voyager



Custom tool

GraphQL의 장점

마이크로서비스와 GraphQL의 연동 이점

- 분리된 REST 서비스 호출을 한번에 처리

Name
☐ user_info
☒ listing_scans?status=completed&includes_relatio...
☐ listing_scans?featured_scans=true&includes_relati...

3 Calls



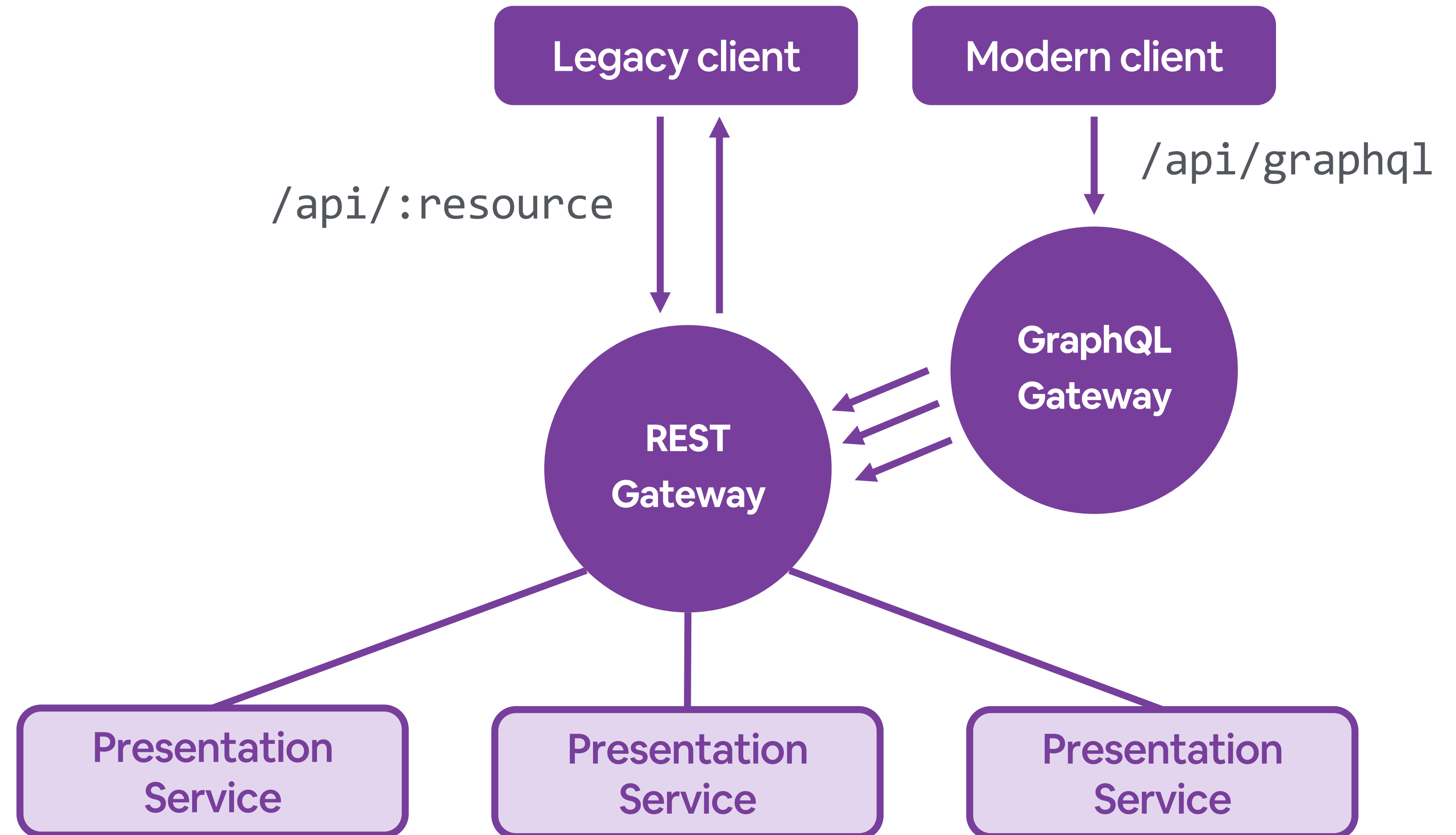
1 Call

```
query getListingScans($offset: Long!, $limit: Long!) {
  getListingScans(
    request: {
      offset: $offset
      limit: $limit
    }
  ) {
    listingScans {
      createdByEmail
      id
      title
      thumbnailUrl
    }
  }
  getListingScansAdmin(request: {offset: $offset, limit: $lim
    listingScans {
      createdByEmail
      title
      thumbnailUrl
    }
  }
  getUserInfo {
    userId
    isAdmin
  }
}
```

```
{
  "data": {
    "getListingScans": {
      "listingScans": [
        {
          "createdByEmail": "Emc",
          "id": "704",
          "title": "Tes",
          "thumbnailUrl": "..."
        },
        {
          "createdByEmail": "Emc",
          "id": "737",
          "title": "Gar",
          "thumbnailUrl": "..."
        },
        {
          "createdByEmail": "Emc",
          "id": "738",
          "title": "Tes",
          "thumbnailUrl": "..."
        }
      ]
    },
    "getListingScansAdmin": {
      "listingScans": [
        {
          "createdByEmail": "Emc",
          "id": "704",
          "title": "Tes",
          "thumbnailUrl": "..."
        },
        {
          "createdByEmail": "Emc",
          "id": "737",
          "title": "Gar",
          "thumbnailUrl": "..."
        },
        {
          "createdByEmail": "Emc",
          "id": "738",
          "title": "Tes",
          "thumbnailUrl": "..."
        }
      ]
    },
    "getUserInfo": {
      "userId": "1",
      "isAdmin": true
    }
  }
}
```

Presentation Service GraphQL

서비스 구조도



Thrift IDL 활용 예제 – GraphQL

Thrift IDL과 GraphQL서비스와의 연동 방법

- 추가 annotation을 활용하여 자동으로 Schema 등록
- graphql-mesh등의 오픈 소스 활용 가능

```
1 struct PdpRequest {
2   1: required i32 productId
3 }
4
5 struct PdpResponse {
6   1: optional i32 id
7   2: optional string name
8   3: optional i32 price
9 }
10
11 service Market {
12   PdpResponse getPdp(1: PdpRequest request)
13   (
14     graphql_operation_type = "query"
15   )
16 }
17
```

The screenshot shows a GraphQL client interface with a dark theme. The main area displays a query and its response. The query is:

```
1 # Write your query or mutation here
2 query {
3   getPdp(request:{productId: 123 }) {
4     id
5     name
6     price
7   }
8 }
```

The response is:

```
{
  "data": {
    "getPdp": {
      "id": 123,
      "name": "에어컨",
      "price": 350000
    }
  }
}
```

On the right side, there is a 'SCHEMA' tab with a 'DOWNLOAD' button. The schema is:

```
type Query {
  #
  getPdp(request: PdpRequestInput): PdpResponse
}

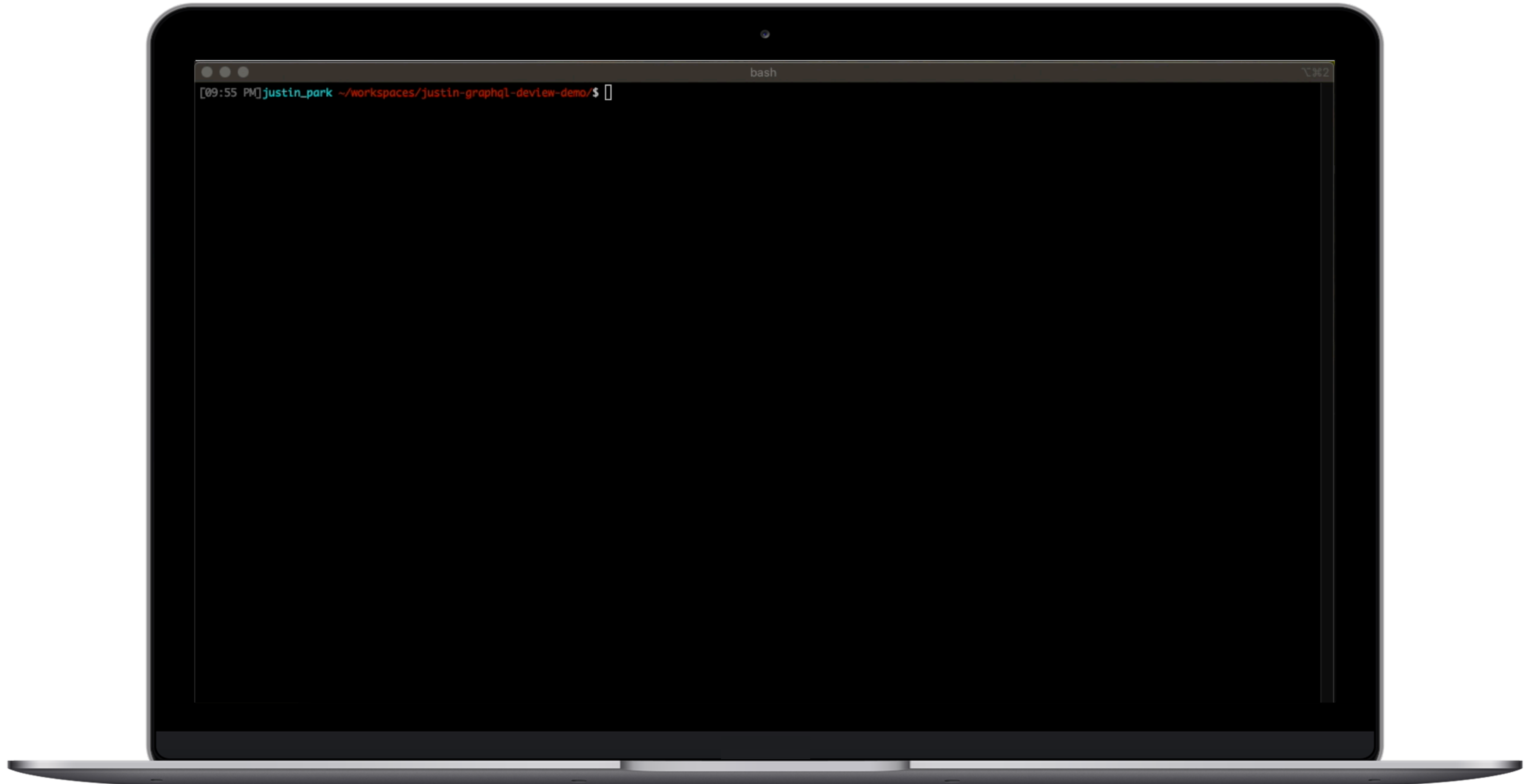
#
type PdpResponse {
  #
  id: Int
  #
  name: String
  #
  price: Int
}

#
input PdpRequestInput {
  #
  productId: Int!
}
```

A purple arrow points from the 'graphql_operation_type = "query"' line in the Thrift IDL code to the 'query' block in the GraphQL query.

Thrift IDL 시연

github.com/justinpark/graphql-devview-demo



Redux 호환성 문제 해결책

컴포넌트 연동 작업

- Container 컴포넌트 재 활용
- Apollo Container에서 GraphQL 데이터를 프로퍼티로 전달

```
1 function PdpRoute(props) {  
2   //...  
3   if (isApolloEnabled()) {  
4     return (  
5       <PdpContainerApollo  
6         productId={productId}  
7         {...props}  
8       />  
9     );  
10  }  
11  return (  
12    <PdpContainerRedux  
13      productId={productId}  
14      {...props}  
15    />  
16  );  
17 }
```

Redux 호환성 문제 해결책

기존 서비스 데이터 연동 작업

- 기존 REST 데이터를 GraphQL 타입에 맞게 변환

```
1 export default function covertPdpData(payload) {
2   return mapToQuery({
3     query: GetPdpQuery,
4     resolvers: {
5       getPdp: () => payload.pdpData,
6       luxuryPicture: {
7         imageUrl: (picture) => picture.image_url,
8       },
9       roomInfo: {
10        roomName: (room) => room.room_name,
11        coverImageUrl: (room) => room.cover_image_url,
12        roomImages: ({ room_images: roomImages }) =>
13          roomImages?.length > 0 ? roomImages : null,
14      },
15    },
16  });
17 }
```


GraphQL 보안 처리 과정

GraphQL 쿼리 인증 처리

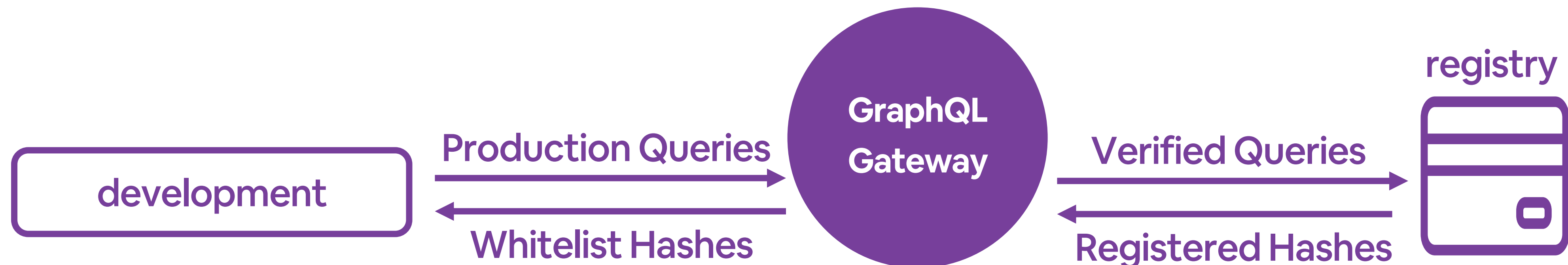
- 요청값에 노출된 쿼리 스키마를 보호
- 개발 과정 중 필요 쿼리만을 미리 등록

```

▼ Request Payload  view source
▼ {operationName: "PdpQuery", variables: {productId: 123},...}
  operationName: "PdpQuery"
  query: "query PdpQuery($productId: Int!) { getPdp(request: {productId: $productId}) { id name p"
  ▼ variables: {productId: 123}
    productId: 123
  
```

```

1 ✓ Loading Apollo config
2 ✓ Fetching current schema
3 ✓ Publishing <service> to Apollo
4
5 id      schema      variant
6 -----
7 abc123  <service>      current
  
```



GraphQL 보안 처리 과정

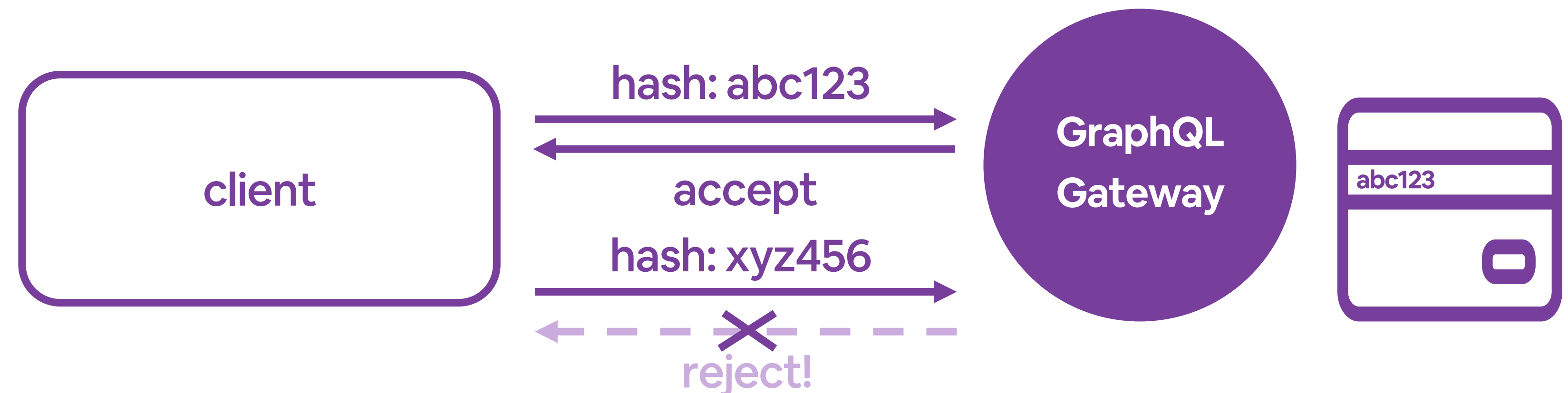
GraphQL 쿼리 인증 처리

- 빌드 과정 중 GraphQL Query를 해쉬값으로 변환
- 등록된 키값과 일치한 쿼리만 실행
- apollo-server-plugin-operation-registry

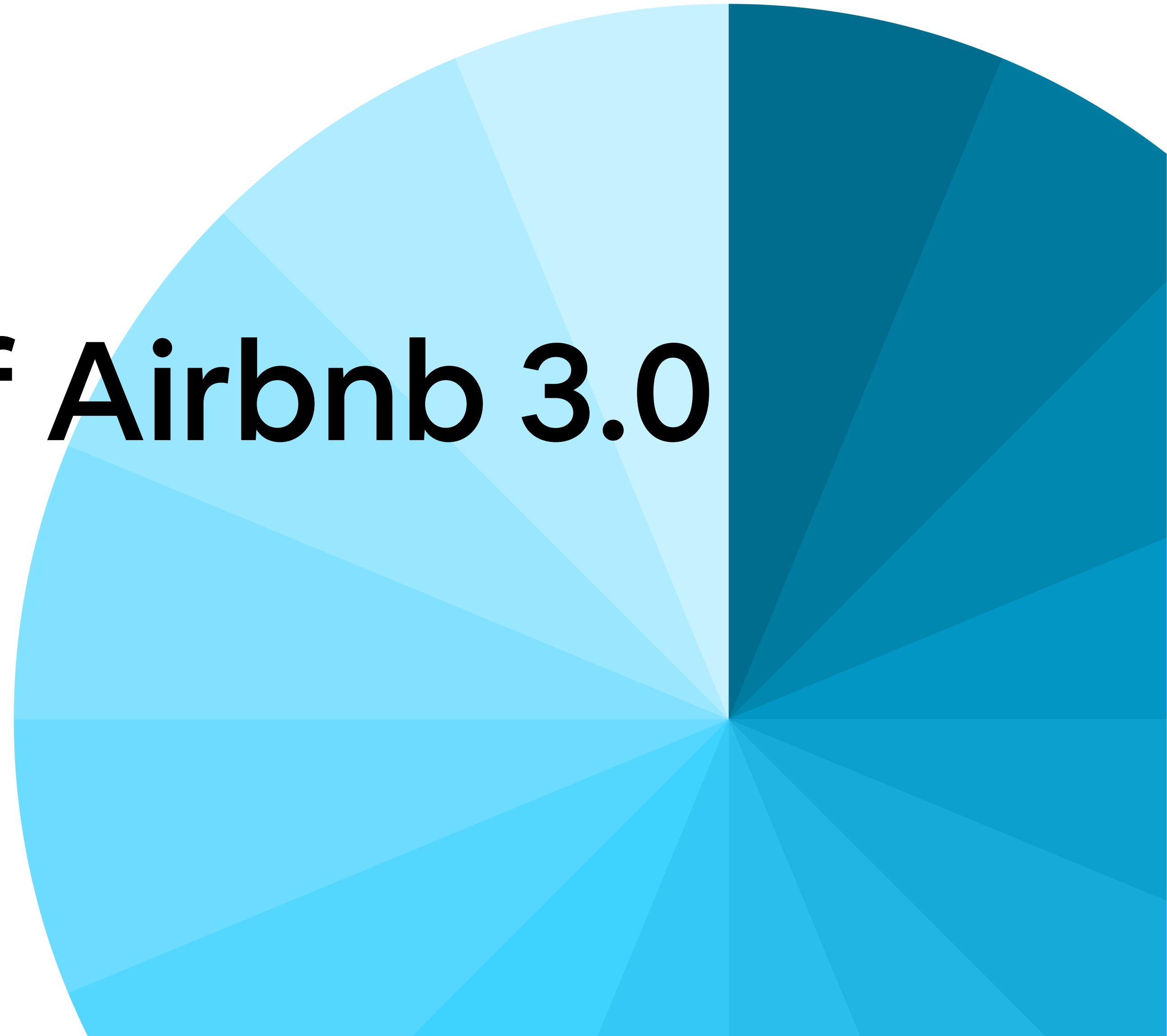
```

▼ Request Payload view source
▼ {operationName: "PdpQuery", variables: {productId: 123},...}
  operationName: "PdpQuery"
  query: "query PdpQuery($productId: Int!) { getPdp(request: {productId: $productId}) { id name price } }"
  ▼ variables: {productId: 123}
    productId: 123

operationName: PdpQuery
currency: USD
variables: {productId: 123}
extensions: {"persistedQuery":{"version":1,"sha256Hash":"7e0a3b470d5c212f395f3b4dcc469f97e0ffd243b60b50dc557a419ef7197b3f"}}
  
```



New era of Airbnb 3.0

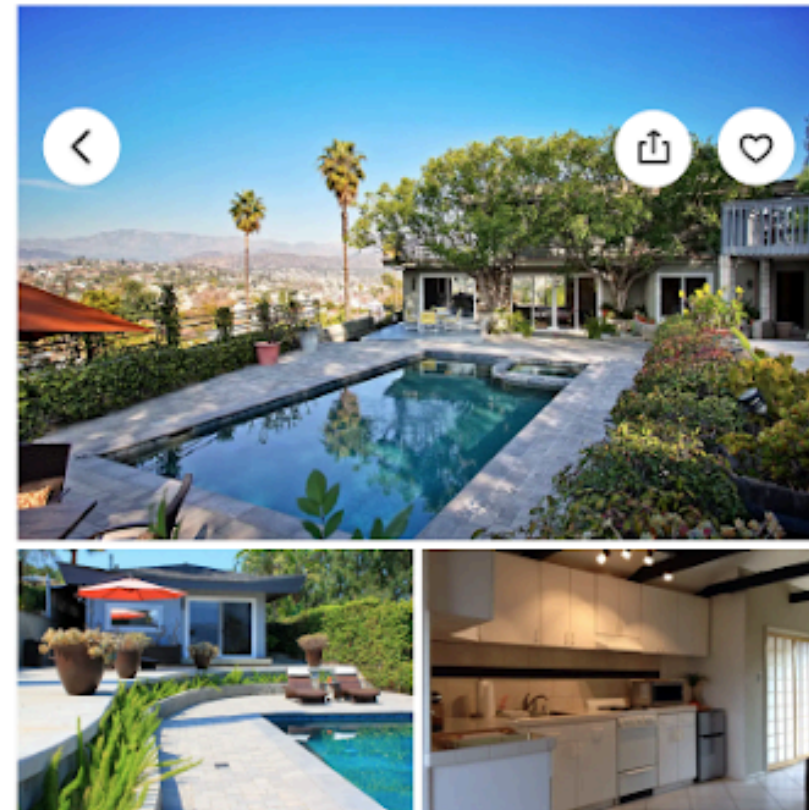


파편화 문제

Marketplace

\$42 ~~\$42~~ **\$34** / night

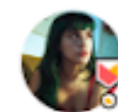
Start booking



Santa Monica guesthouse with pool and garden

★ 4.92 (156) · [Los Angeles, California](#)

Entire home
hosted by Novak



3 guests · 1 bedroom · 2 beds · 1 bath



Sparkling clean

As reported by 90% of recent guests.



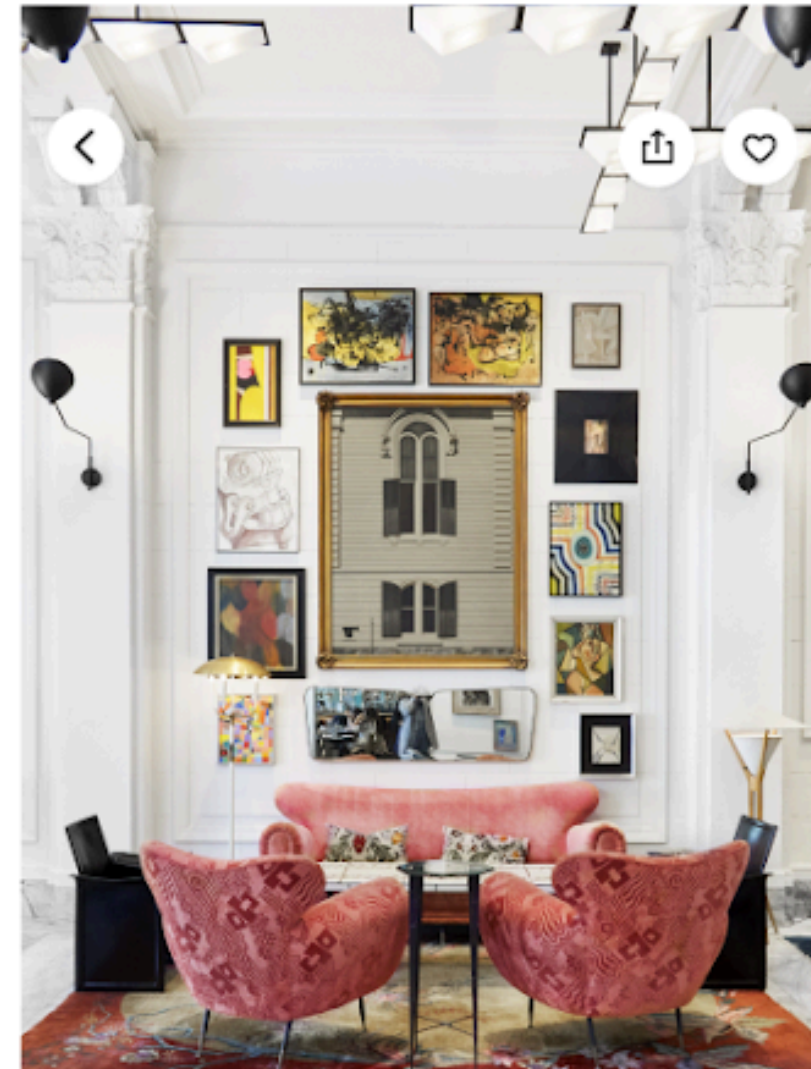
Great location

90% of recent guests gave the location

Hotels

From **\$380** / night

Select room



Proper Hotel

★ 4.92 (15) · [Mid-Market, San Francisco](#)

This is a boutique hotel

131 rooms · Onsite restaurants



Great rooftop bar

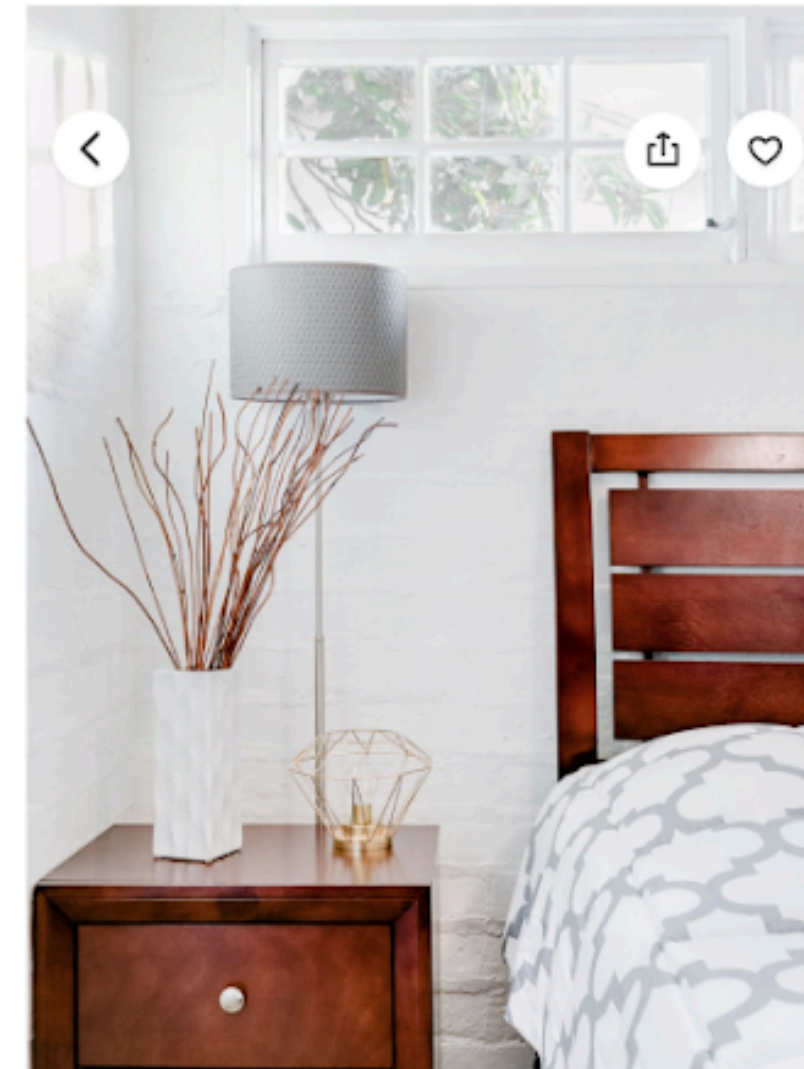
15 recent guests were wild about the

Plus

\$95 / night

PLUS Verified

Check availability



Plus

Soak Up the LA Feel from a
Private Pool Home

★ 4.92 (156) · [Los Angeles, California](#)

Entire home
hosted by Novak



3 guests · 1 bedroom · 2 beds · 1 bath

Luxe

\$3,500 / night

LUXE Verified

Check availability



Luxe

Villa Kandara

Luxury stay in [Los Angeles, California](#)

파편화 문제

유사한 기능을 팀별로 반복적으로 제작하는 문제 발생

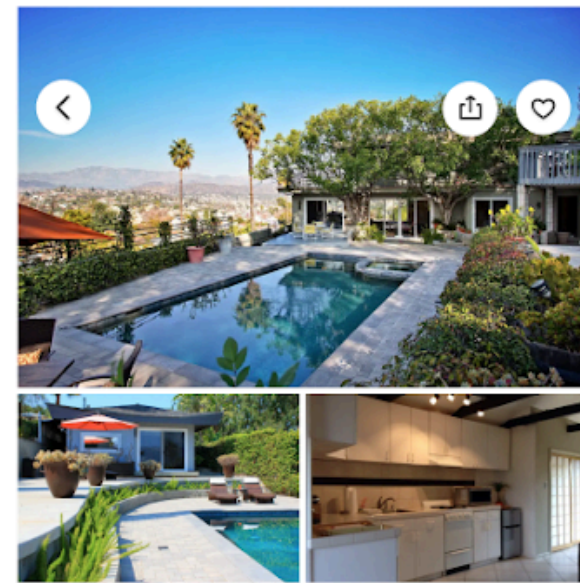
xt
nb
uct

flights

Marketplace

\$42 \$34 / night

Start booking



Santa Monica guesthouse with pool and garden

★ 4.92 (156) · Los Angeles, California

Entire home
hosted by Novak



3 guests · 1 bedroom · 2 beds · 1 bath



Sparkling clean

As reported by 90% of recent guests.



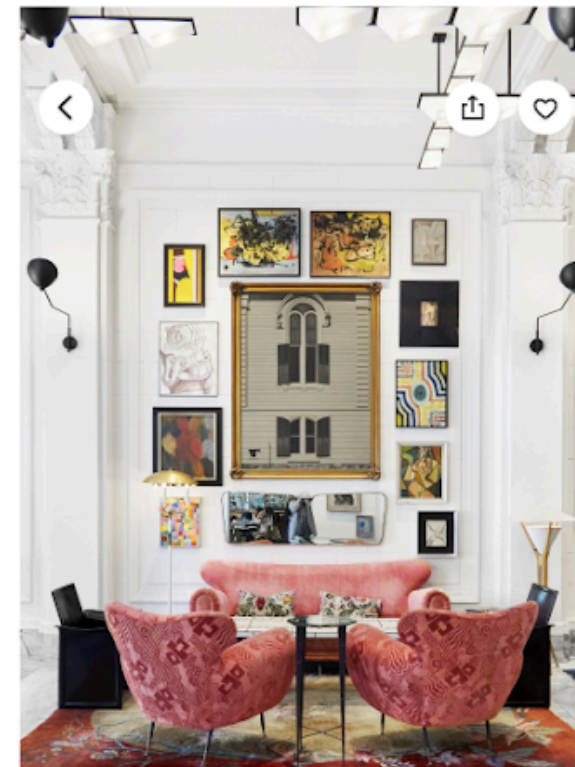
Great location

90% of recent guests gave the location

Hotels

From \$380 / night

Select room



Proper Hotel

★ 4.92 (15) · Mid-Market, San Francisco

This is a boutique hotel

131 rooms · Onsite restaurants



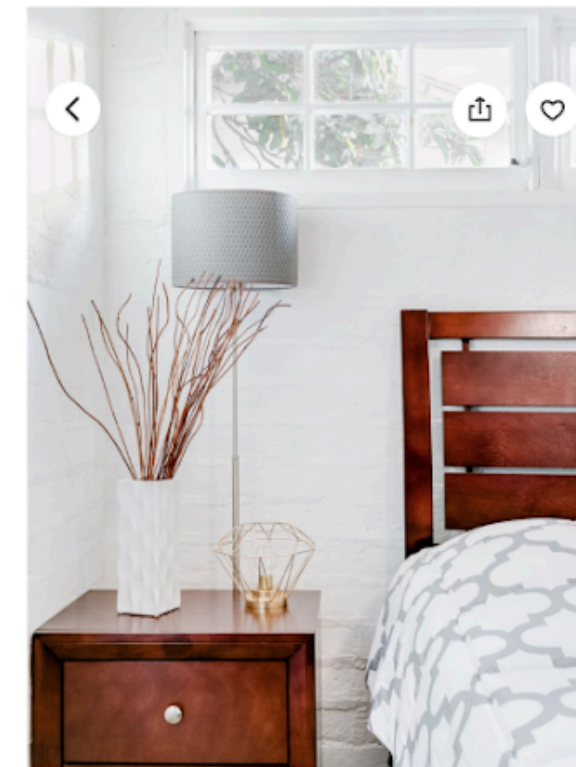
Great rooftop bar

15 recent guests were wild about the

Plus

\$95 / night
PLUS Verified

Check availability



Plus

Soak Up the LA Feel from a Private Pool Home

★ 4.92 (156) · Los Angeles, California

Entire home
hosted by Novak



3 guests · 1 bedroom · 2 beds · 1 bath

Luxe

\$3,500 / night
LUXE Verified

Check availability



Luxe

Villa Kandara

Luxury stay in Los Angeles, California

trip

ne
air
pro

복잡한 데이터 구조

Confirm and pay



This is a rare find. Shilpi's place is usually booked.

Pay with



Credit or debit card



Apply \$250.43 gift credit



[Apply coupon](#)

How do you want to pay for this trip?



Pay in full

\$1,792.18



Pay less up front

\$896.09

The rest of \$896.09, due Mar 15.
No extra fees. [Learn more](#)

Cancellation policy: Moderate

Cancel up to 7 days before check in and get a 50% refund (minus service fees). Cancel within 7 days of your trip and the reservation is non-refundable. Service fees are refunded when cancellation happens before check in and within 48 hours of booking.

I agree to the [House Rules](#), [Cancellation Policy](#), and to the [Guest Refund Policy](#). I also agree to pay the total amount shown, which includes Service Fees.



Confirm and pay

Sunny Modern Home in The Haight district

Entire home in San Francisco



Mar 23, 2018



Mar 28, 2018



2 guests, 1 infant

\$174.08 x 11 nights

\$1,914.88

Cleaning fee

\$93.47

Service fee

\$199.15

Total

\$2,207.50

Due now (USD)

\$1,103.75

Due on Oct 26, 2018

\$1,103.75

Moderate — free cancellation

Cancel within 48 hours of booking to
get a full refund. [More details](#)



복잡한 데이터 구조

Home demand service

Coupon service

Reservation service

Payments service

Confirm and pay

 This is a rare find. Shilpi's place is usually booked.

Pay with



 Credit or debit card

 Apply \$250.43 gift credit

 Apply coupon

How do you want to pay for this trip?

☒ Pay in full \$1,792.18

☐ Pay less up front \$896.09

The rest of \$896.09, due Mar 15.
No extra fees. [Learn more](#)

Cancellation policy: Moderate

Cancel up to 7 days before check in and get a 50% refund (minus service fees). Cancel within 7 days of your trip and the reservation is non-refundable. Service fees are refunded when cancellation happens before check in and within 48 hours of booking.

I agree to the [House Rules](#), [Cancellation Policy](#), and to the [Guest Refund Policy](#). I also agree to pay the total amount shown, which includes Service Fees.

 Confirm and pay

Sunny Modern Home in
The Haight district

Entire home in San Francisco



 Mar 23, 2018 → Mar 28, 2018

 2 guests, 1 infant

\$174.08 x 11 nights \$1,914.88

Cleaning fee \$93.47

Service fee \$199.15

Total \$2,207.50

Due now (USD) \$1,103.75

Due on Oct 26, 2018 \$1,103.75

Moderate — free cancellation

Cancel within 48 hours of booking to
get a full refund. [More details](#)



Home service

Reservation service

Pricing service

Cancellation service

복잡한 데이터 구조

해결되지 않은 문제들

- 인터페이스만 GraphQL로 변경
- 데이터 결과는 동일
- 학습 장벽, 코드양만 증가

```

1 query {
2   pdpBooking(request: $request) {
3     ...PdpPricingQuote
4   }
5 }

```

```

1 fragment PdpPricingQuote on PdpBookingResponse {
2   available
3   check_in: checkIn
4   check_out: checkOut
5   nights
6   rate_type: rateType
7   should_show_from_label: shouldShowFromLabel
8   p3_message_data: p3MessageData {
9     ...UncMessageData
10  }
11  base_price_breakdown: basePriceBreakdown {
12    ...DisplayRate
13  }
14  extra_guest_fee: extraGuestFee {
15    ...DisplayRate
16  }
17  price {
18    ...Price
19  }
20  payment_plan_preview_data: paymentPlanPreviewData {
21    ...PaymentPlanPreviewData
22  }
23  secondary_display_rate_data: secondaryDisplayRateData {
24    display_strategy: displayStrategy
25    localized_secondary_price_string: localizedSecondaryPriceString
26  }
27  p3_display_rate: p3DisplayRate {
28    amount
29    currency
30    amount_formatted: amountFormatted
31  }
32  localized_unavailability_message_position_string:
33    localizedUnavailabilityMessagePositionString
34  guest_details: guestDetails {
35    number_of_adults: numberOfAdults
36    number_of_children: numberOfChildren
37    number_of_infants: numberOfInfants
38  }
39  p3_cancellation_section: p3CancellationSection {
40    ...P3CancellationSection
41  }
42  highlights_section: highlightsSection {
43    highlights {
44      title
45      subtitle
46      icon
47    }
48  }
49  shouldDefaultBizToggleForCovid19
50 }

```

Reservation service

Messaging service

Pricing service

Reservation service

Cancellation service

Home demand service

Business travel service

복잡한 데이터 구조

해결되지 않은 문제들

- 기술 부채: 데이터 복잡성은 그대로 유지

```

▼ scheduled_trip: {id: 29108545, starts_at: "2020-11-15 14:30:00.0",
  base_price_string: "$12"
  booking_closes_on: "2020-11-14T09:30:00-05:00"
  computed_shared_booking_type: "private_booking_allowed"
  earliest_departure_date: "2020-11-16"
  end_date: "2020-11-15"
  id: 29108545
  is_hatch: false
  language_codes: ["en"]
  max_guests: 5
  min_nights_in_market: 1
  num_days: 1
  num_guests: 0
  policy_items: [{title: "Language", description: "This experience
    ▼0: {title: "Language", description: "This experience is hosted
      description: "This experience is hosted in English."
      title: "Language"
      type: "language_requirements"
    ▼1: {title: "Health and safety guidelines", type: "safety_messa
      html_description: "In light of COVID-19, Airbnb has created
      title: "Health and safety guidelines"
      type: "safety_message"
    price_per_guest: 12
    private_booking_price: 20
    private_booking_price_string: "$20"
    promotional_price_string: "$12"
    remaining_capacity: 5
    remaining_hatch_capacity: 0
  scheduled_experiences: [...]
    ▼0: {raw_starts_at: "2020-11-15 14:30:00.0", raw_ends_at: "2020
      date_display_string: "9:30 AM - 12:00 PM"
      date_time_display_string: "Sun, Nov 15 · 9:30 AM - 12:00 PM"
      ends_at: "2020-11-15T17:00:00Z"
      ends_at_localized: "2020-11-15T12:00:00-05:00"
      experience_id: 1380054
  
```

Who's coming?

Book just for you and your group

You can book this experience for up to 5 guests starting at \$20.

Number of guests

1 guest

Guest details

The info entered will be used to add people to this reservation.

Ho Joon Park

Payment method

Pay with

VISA AMEX DISCOVER PayPal G Pay

Sun, Nov 15

9:30 AM - 12:00 PM

\$12.00 x 1 guest

\$12.00

Credits

-\$12.00

Total (USD)

\$0.00

Cancellation policy

Cancel within 24 hours of booking for a **full refund** under the [experiences cancellation policy](#). Cancellations related to coronavirus (COVID-19) are addressed under our [extenuating circumstances policy](#). For reservations made after March 14, coronavirus (COVID-19) cancellations will not qualify as an extenuating circumstance. Please review official advisories and take the necessary precautions.

Review guest requirements

Guests ages 18 and up can attend.

Guests should bring water and possibly a light snack. Wear sunscreen and/or a hat and sunglasses.

추가 작업만 늘어날 뿐...

특정 항목만 추출하는 것은 좋으나 호출 작업이 늘어남

```
1 const bookingResponse = Api.get('/api/pdp', { productId });
```

VS

```
const { data } = useQuery(query, {
  variables: { productId },
});
```

+

```
1 fragment PdpPricingQuote on PdpBookingResponse {
2   available
3   check_in: checkIn
4   check_out: checkOut
5   nights
6   rate_type: rateType
7   should_show_from_label: shouldShowFromLabel
8   p3_message_data: p3MessageData {
9     ...UncMessageData
10  }
11   base_price_breakdown: basePriceBreakdown {
12     ...DisplayRate
13   }
14   extra_guest_fee: extraGuestFee {
15     ...DisplayRate
16   }
17   price {
18     ...Price
19   }
20   payment_plan_preview_data: paymentPlanPreviewData {
21     ...PaymentPlanPreviewData
22   }
23   secondary_display_rate_data: secondaryDisplayRateData {
24     display_strategy: displayStrategy
25     localized_secondary_price_string: localizedSecondaryPriceString
26   }
27   p3_display_rate: p3DisplayRate {
28     amount
29     currency
30     amount_formatted: amountFormatted
31   }
32   localized_unavailability_message_position_string:
33     localizedUnavailabilityMessagePositionString
34   guest_details: guestDetails {
35     number_of_adults: numberOfAdults
36     number_of_children: numberOfChildren
37     number_of_infants: numberOfInfants
38   }
39   p3_cancellation_section: p3CancellationSection {
40     ...P3CancellationSection
41   }
42   highlights_section: highlightsSection {
43     highlights {
44       title
45       subtitle
46       icon
47     }
48   }
49   shouldDefaultBizToggleForCovid19
50 }
```


Thinking in GraphQL way

서버 기반 UI (Server-driven UI)

서버에 정의된 메타데이터(metadata)를 기반으로
화면을 구성하는 방법

```
{  
+ title: 'Fees paid at property',  
  subsections:  
+   title: 'Resort fee',  
}]  
}
```

Hotel San Cristóbal Baja

Boutique hotel

Property details

Photos

Spaces and amenities

Policies

Fees

Local laws and taxes

Fees paid at property

Add extra fees that guests will have to pay during their stay. These fees won't be charged on booking, so confirm them here will help guests know what to expect.

Resort fee

\$40 per night

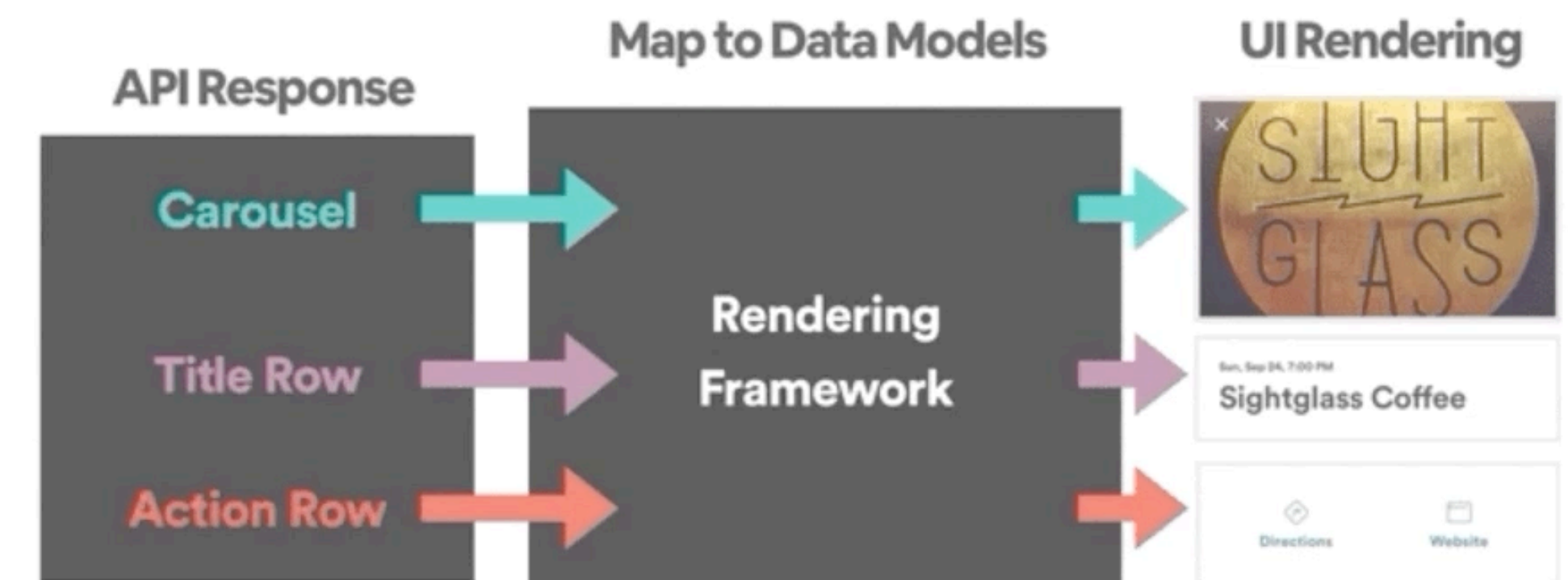
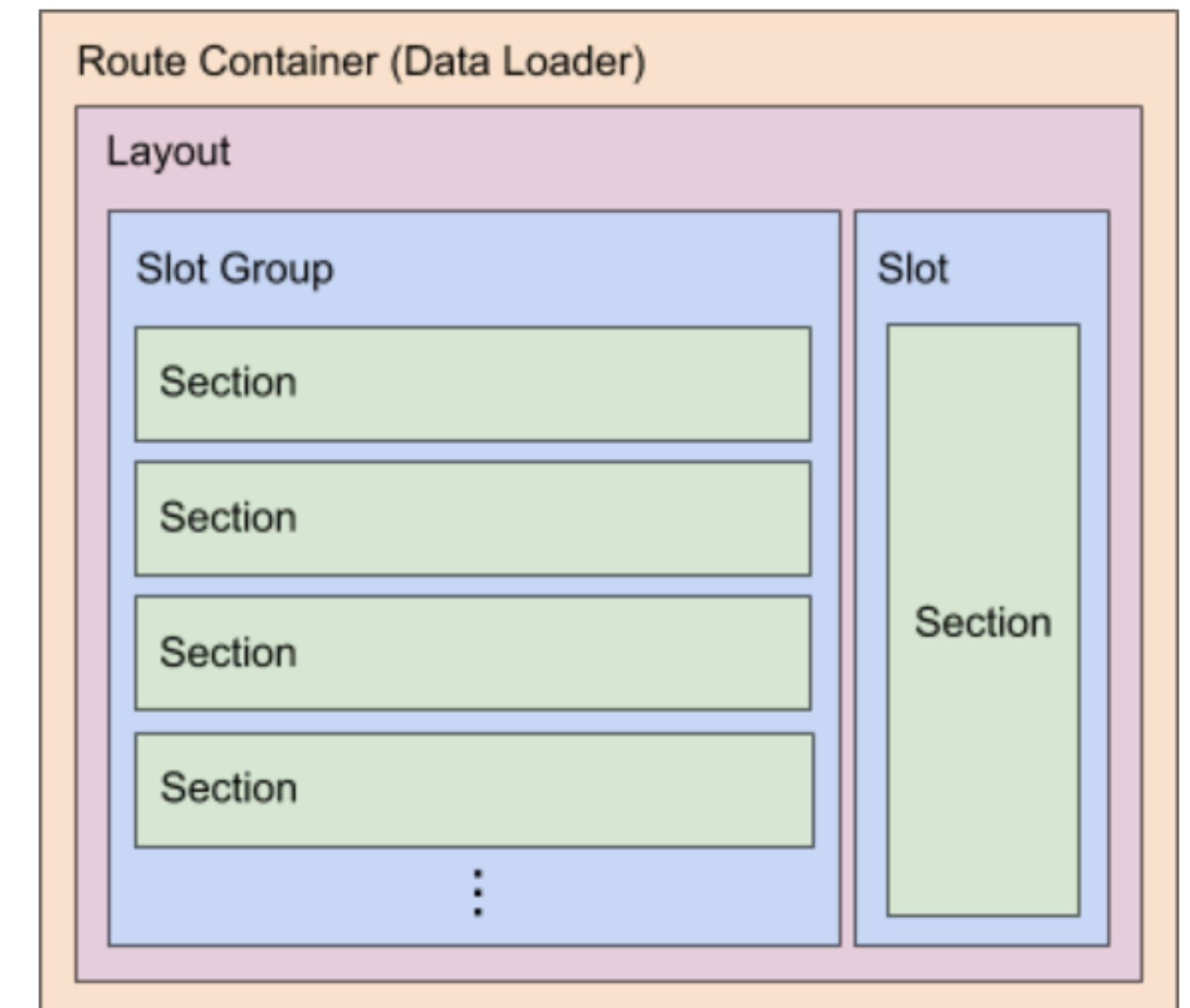
Currency

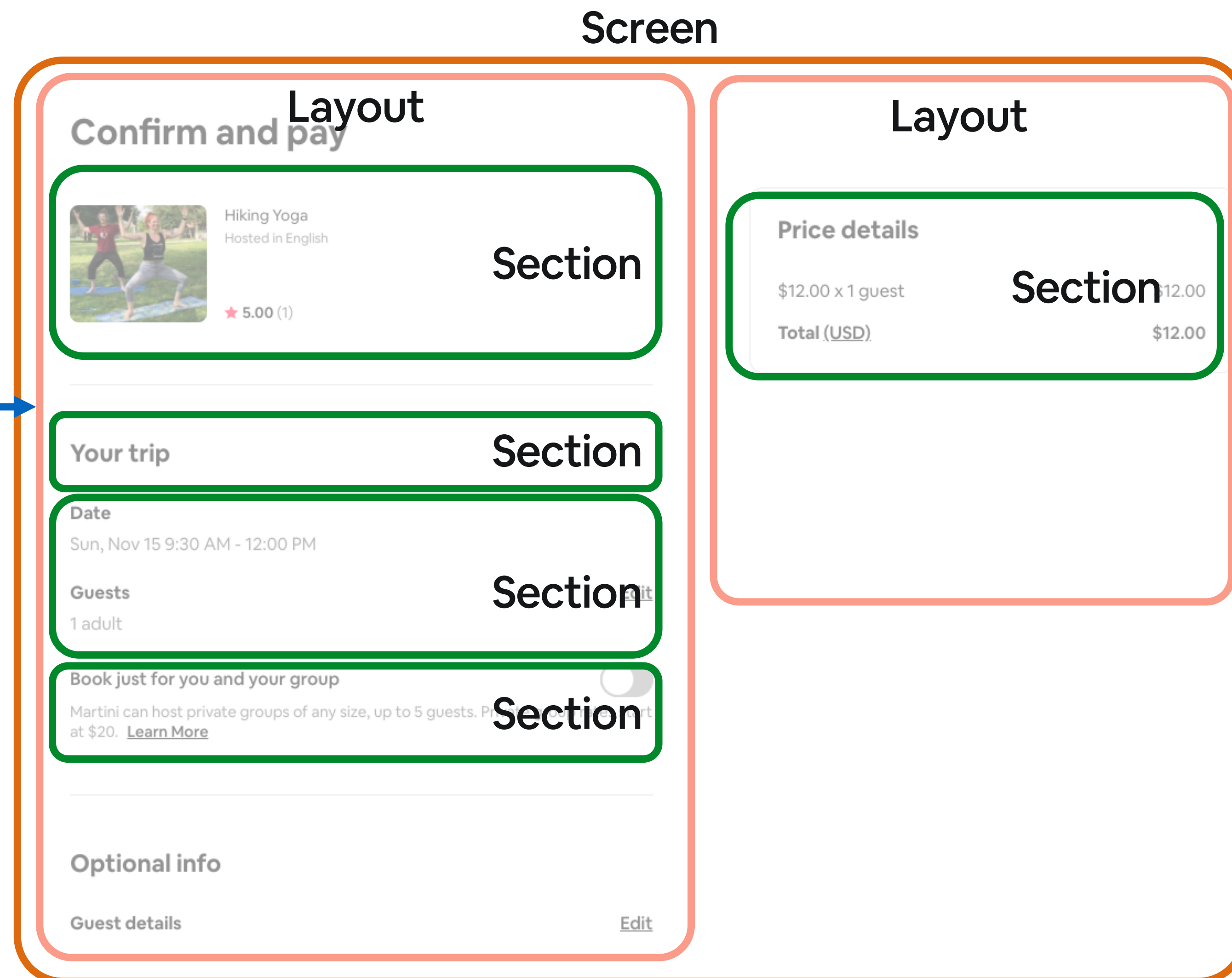
USD

서버 기반 UI (Server-driven UI)

서버 기반 UI의 장점

- 하나의 정의된 메타데이터로 동일한 화면 구성 가능
- 재배포를 통해 여러 플랫폼에서 재사용 가능





UI 구성요소: 메타데이터

Context

- 컨텍스트를 통해 메타데이터와 컴포넌트 정보를 관리

```
1 function ClientApp({ data, error, loading }) {  
2   const { metadata } = data;  
3  
4   return (  
5     <AppFramework  
6       registry={registry}  
7       metadata={metadata}  
8     />  
9   );  
10 }
```

```
1 function AppFramework({ registry, metadata, children }) {  
2   const { sections, screen } = metadata;  
3   const { layouts } = screen;  
4   const sections = new Map(sections.map((section) => [section.id, section]));  
5   const layouts = new Map(...);  
6   const context = {  
7     sections,  
8     layouts,  
9     registry,  
10  };  
11  
12  return (  
13    <Context.Provider value={context}>  
14      {children}  
15    </Context>  
16  );  
17 }
```


UI 구성요소: 컴포넌트 맞춤 설정

Registry

- 메타데이터 정보와 컴포넌트를 연결
- 서버 사이드 랜더링과
지연 로딩을 통해 최적화 가능

```
1 import Title from './Title';
2
3 // An example loader for an async component
4 const locationLoader = () =>
5   import(
6     /* webpackChunkName: "location" */
7     './Location'
8   );
9
10 // For Server-side rendering
11 if (typeof window === 'undefined') {
12   locationLoader.value = require('./Location');
13 }
14
15 export const registry = {
16   // A regular component
17   'titleSection': {
18     component: Title,
19   },
20
21   // An async component with a loader
22   'locationSection': {
23     loader: locationLoader,
24   },
25 };
```

UI 구성요소: 화면 구성

Screen component

- 메타데이터의 레이아웃을 실제 화면에 배치하는 과정
- 키 값으로 분리하여 여러 플랫폼에 적용 가능

```
1 function PageScreen() {  
2   return (  
3     <>  
4       <Column md={8}>  
5         <Layout id="MAIN" />  
6       </Column>  
7       <Column md={4}>  
8         <Layout id="SIDEBAR" />  
9       </Column>  
10    </>  
11  );  
12 }
```

UI 구성요소: 컴포넌트 배치 및 출력

Layout

- 메타데이터 컨텍스트를 참조하여 registry에 등록된 컴포넌트를 출력하는 과정

Section

- 지연 로딩 과정 처리

```
1 export default function Layout({ id }) {  
2   const { layout } = useAppFrameworkContext();  
3   const activeSection = layout.get(id);  
4  
5   return (  
6     <>  
7       {activeSection.subsectionIds.map((sectionId) => (  
8         <Section key={sectionId} id={sectionId} />  
9       ))}  
10    </>  
11  );  
12 }
```

```
1 export default function Section({ id }) {  
2   const { registry, sections } = useAppFrameworkContext();  
3   const config = sections.get(id);  
4   const { component, loader } = registry[id];  
5   const [Component, setComponent] = useState(component);  
6  
7   useEffect(() => {  
8     if (loader) {  
9       loader().then((Comp) => setComponent(Comp));  
10    }  
11  }, [loader]);  
12  
13  return Component && <Component id={id} {...config} />;  
14 }
```


GraphQL 쿼리 구조

Union type

- 유니언 구조를 사용하여
한 곳에 모든 섹션 정보를 통합

```

1 struct Screen {
2   1: optional string id
3   2: optional set<Layout> layouts
4 }
5
6 struct SectionContainer {
7   1: optional string id
8   2: optional SectionComponentType sectionComponentType
9   3: optional AvailableSection section
10 }
11
12 union AvailableSection {
13   1: optional Title
14   2: optional Banner
15 }
16
17 struct Banner {
18   1: optional Icon icon
19   2: optional string title
20   3: optional string subtitle
21   4: optional string ctaCopy
22   5: optional string ctaUrl
23 }
24
25 struct Title {
26   1: optional string title
27   2: optional string subtitle
28 }

```

```

1 fragment LayoutFragment on Layout {
2   id
3   sections {
4     sectionId
5   }
6 }
7
8 fragment ScreenFragment on Screen {
9   id
10  name
11  layout {
12    ...LayoutFragment
13  }
14 }
15
16 fragment CheckoutPageFragment on App {
17   sections {
18     ...SectionFragments
19   }
20   screens {
21     ...ScreenContainerFragment
22   }
23 }
24
25 query CheckoutPageQuery($id: ID!) {
26   presentation {
27     checkout(request: { productId: $id, layouts: [MOWEB, DESKTOP] }) {
28       ...CheckoutPageFragment
29     }
30   }
31 }

```

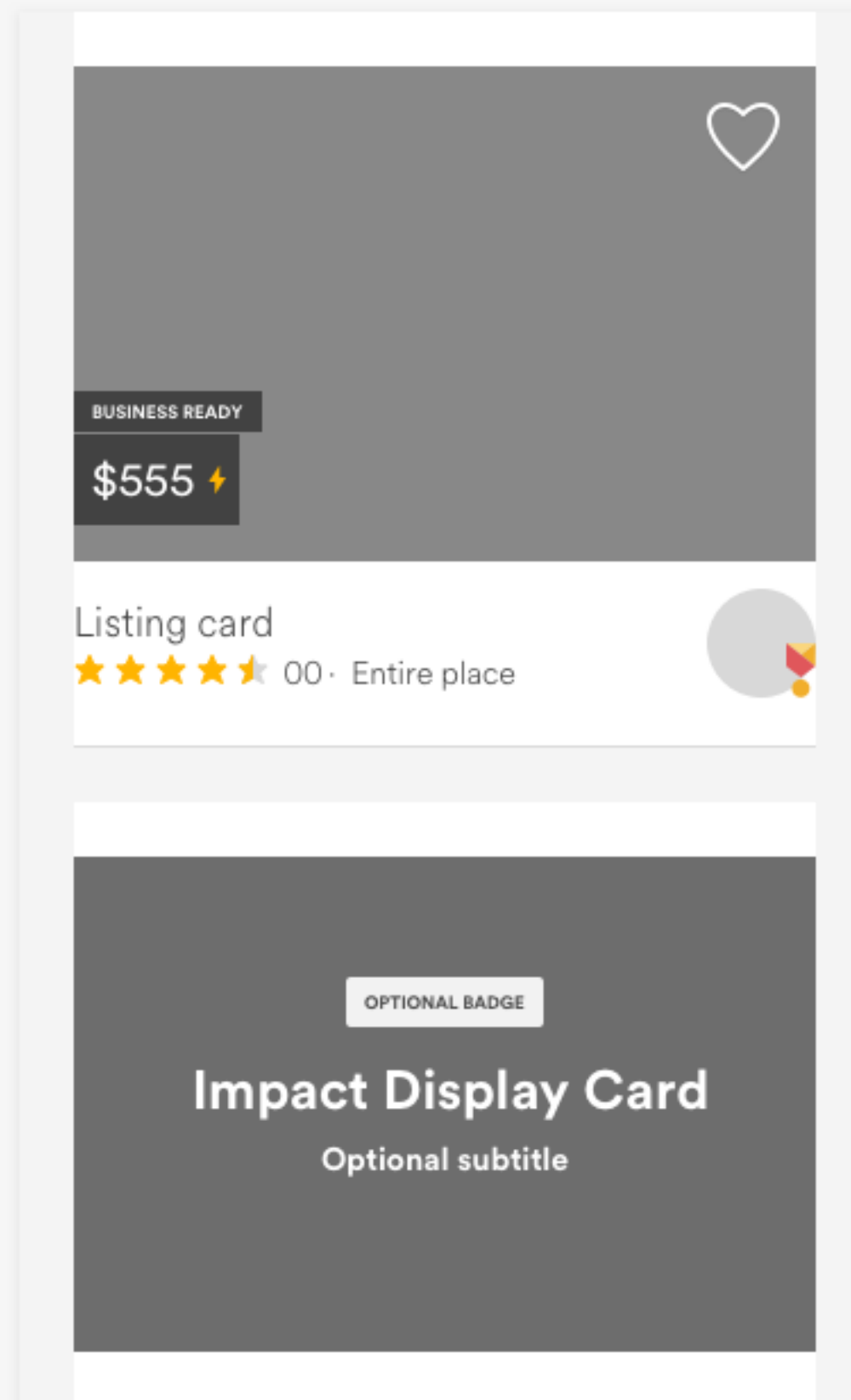
```

1 fragment SectionFragments on Section {
2   id
3   section {
4     ...TitleSectionFragment
5     ...ErrorMessageSectionFragment
6     ...PriceQuoteSectionFragment
7   }
8 }
9

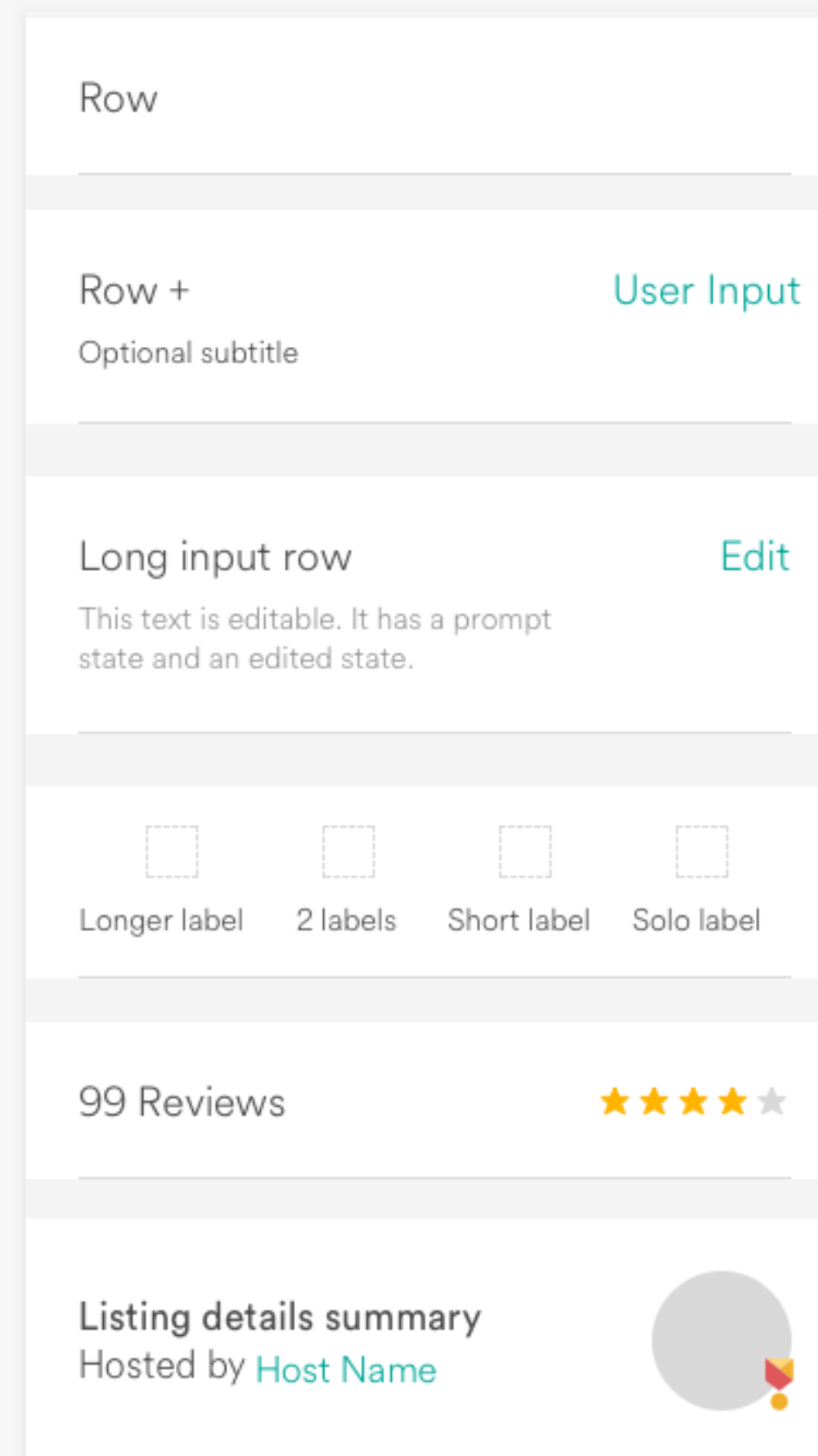
```

디자인 시스템을 넘어서

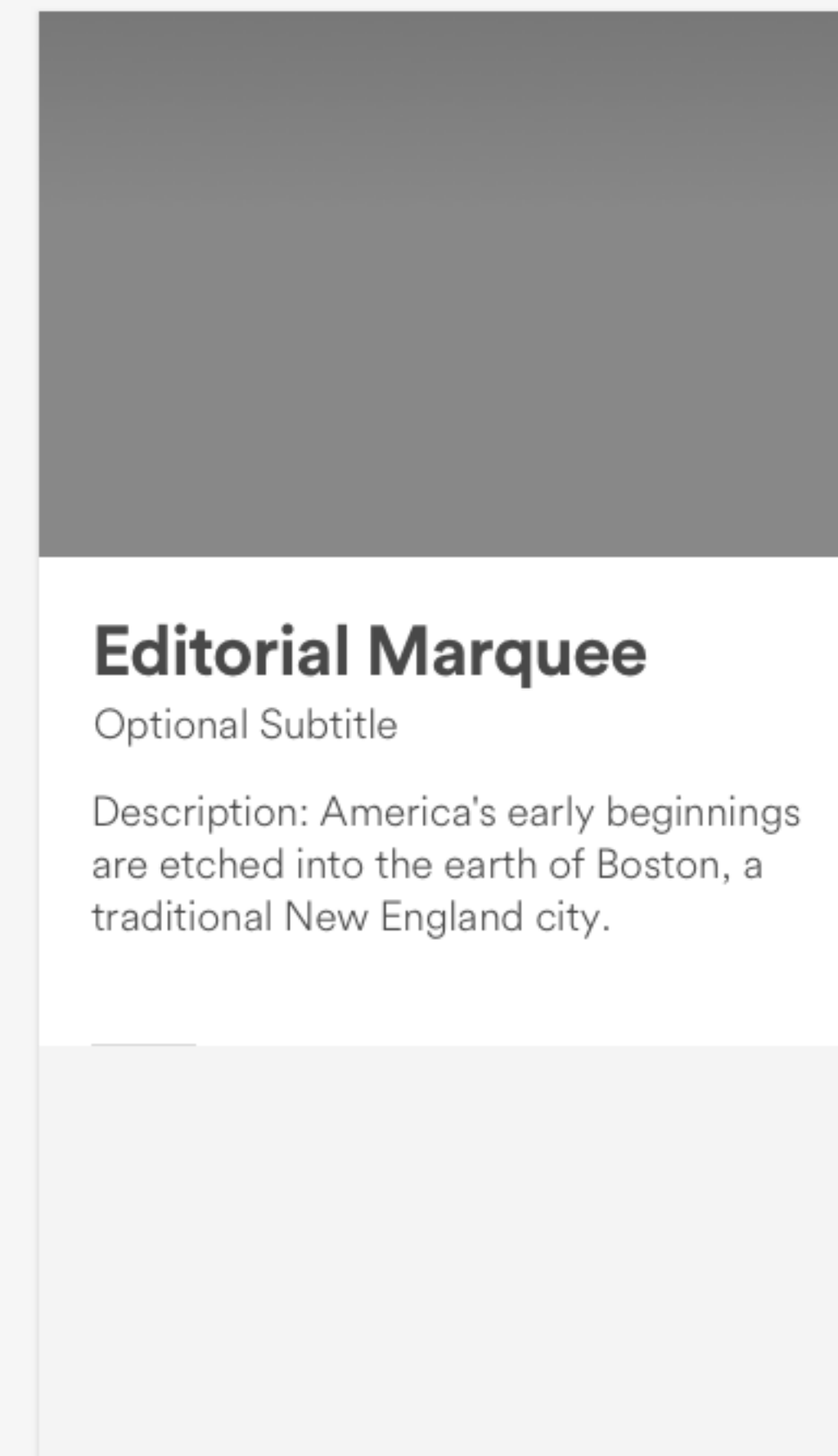
Cards



Rows



Editorial Marquee



섹션 컴포넌트 구조

섹션 컴포넌트란?

- Section 위치에 출력되는 컴포넌트
- 실제 화면 일부분의
데이터를 정의하는 과정
- 쿼리 전체가 아닌
컴포넌트에 필요한 조각만
분리하여 구성 가능

```
1 import React from 'react';
2 import DlsTitleRow from ':dls-web/components/TitleRow';
3 import { TitleSectionFragment } from '.__generated__/TitleSectionFragment';
4
5 type Props = Omit<TitleSectionFragment, '__typename'>;
6
7 export default function TitleSection({
8   section,
9 }: Props) {
10   const { title, subtitle } = section;
11
12   return (
13     <DlsTitleRow
14       label={title}
15       subtitle={subtitle || undefined}
16     />
17   );
18 }
19
```

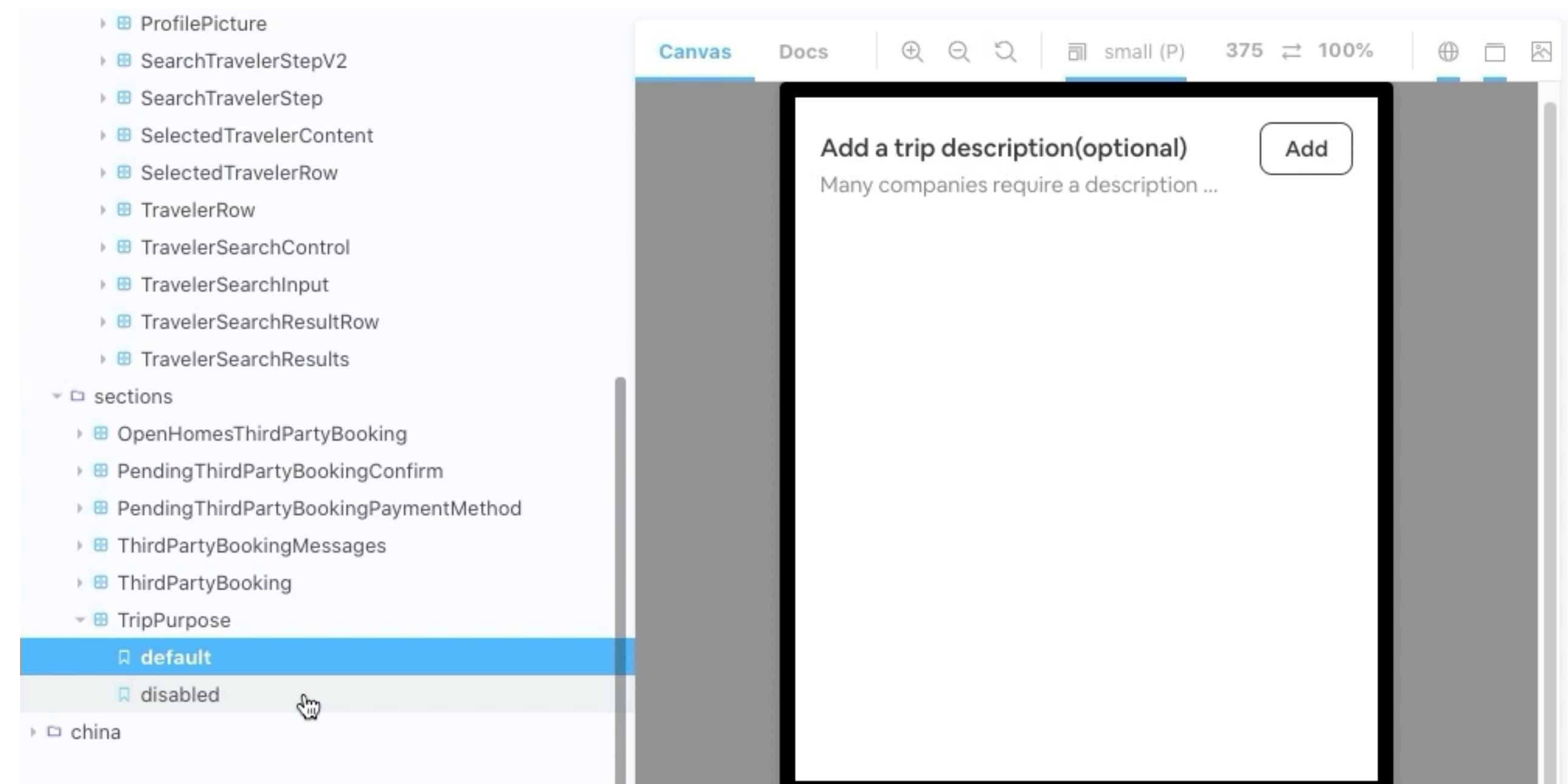
```
1 > TitleSection
2   index.tsx
3   TitleSectionFragment.ts
4
5 > ErrorMessageSection
6 > PriceQuoteSection
```

```
1 import gql from '@apollo/client';
2
3 export default gql`
4   fragment TitleSectionFragment on TitleSection {
5     title
6     subtitle
7   }
8 `;
9
```

스토리북

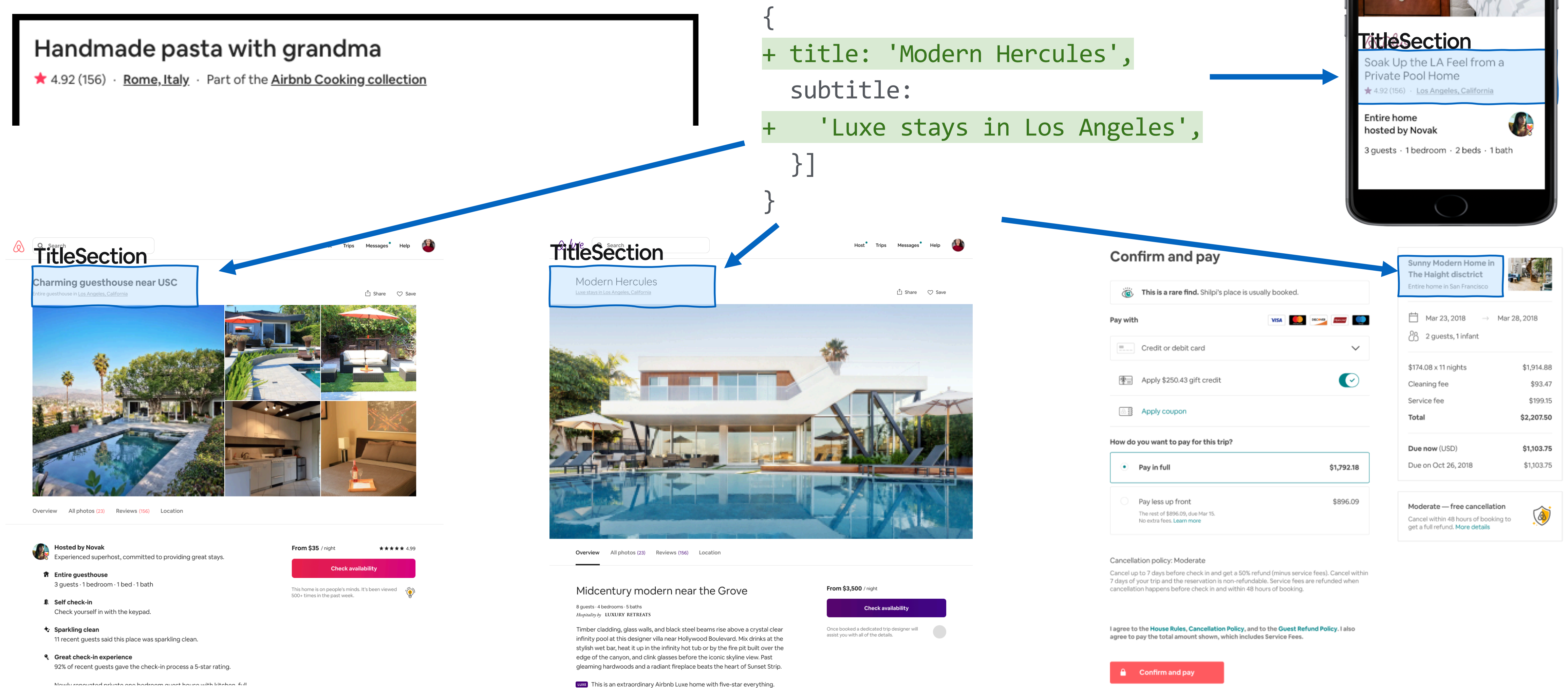
Storybook을 활용한 개발

- 정의된 데이터를 조합하여
출력 가능한 여러 조건을
미리 검증



원소스 멀티 유즈(One Source Multi Use)

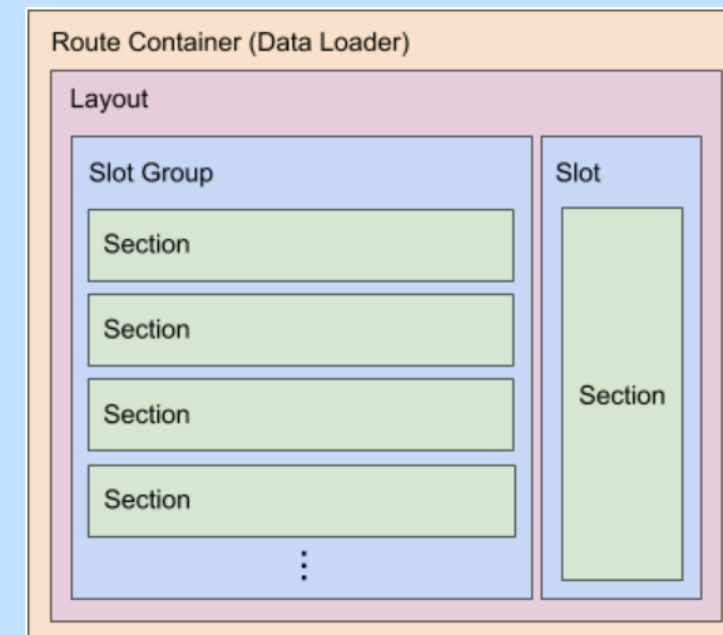
N DEVIEW
2020



완성된 페이지

Screen

```
"screens": [
  {
    "layout": "MAIN",
    "sectionDetails": [
      {
        "sectionId": "DESKTOP_TITLE"
      },
      {
        "sectionId": "CANCELLATION_POLICY"
      }
    ]
  },
  {
    "layout": "SIDEBAR",
    "sectionDetails": [
      {
        "sectionId": "PRICE_TITLE"
      }
    ]
  }
]
```



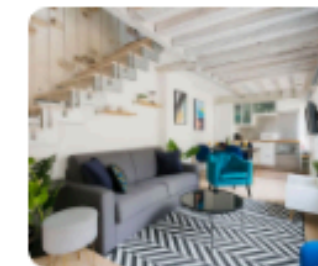
```
"sections": [
  {
    "id": "DESKTOP_TITLE",
    "sectionId": "DESKTOP_TITLE",
    "sectionComponentType": "DESKTOP_PAGE_TITLE",
    "title": "Devview 2020 Demo"
  },
  {
    "id": "CANCELLATION_POLICY",
    "sectionId": "CANCELLATION_POLICY",
    "sectionComponentType": "CANCELLATION_POLICY_DEFAULT",
    "title": "Make sure this host's cancellation policy works for you.",
    "subtitle": "For reservations made after March 14, COVID-19 will ..."
  },
  {
    "id": "PRICE_TITLE",
    "sectionId": "PRICE_TITLE",
    "sectionComponentType": "DESKTOP_PAGE_TITLE",
    "title": "Price details"
  }
]
```

Section

Layout

Section

Devview 2020 Demo



Photograph Milan after dark with locals and other people from Italy
Hosted in English, Italian

★ 4.98 (83)

Make sure this host's cancellation policy works for you.
For reservations made after March 14, COVID-19 will not qualify as an extenuating circumstance.



Your trip

Dates

Dec 16, 2019, 12:30PM

[Edit](#)

Guests

2 adults

[Edit](#)

Book as private group

This experience can be booked privately for up to 20 guests [Learn More](#)



Layout

Price details

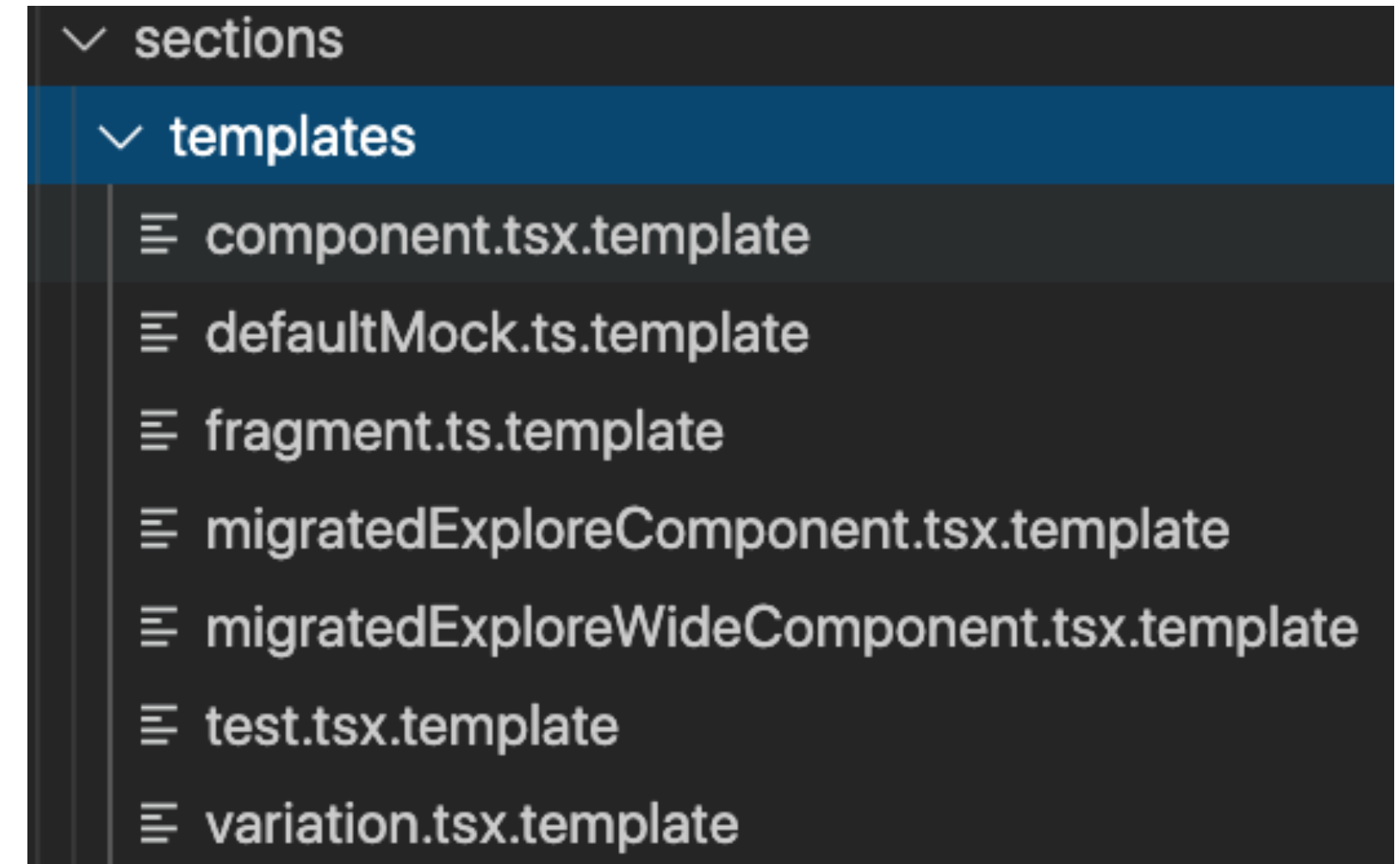
\$45.38 x 2 nights	\$90.76
<u>Cleaning fee</u>	\$64.82
<u>Service fee</u>	\$0.00
<u>Occupancy taxes and fees</u>	\$5.67
<u>Total (USD)</u>	\$161.25



Future of Airbnb 4.0

CLI툴을 활용한 개발 과정 자동화

- GraphQL로 전달받은 스키마 정보 활용
- boilerplate 코드들 자동으로 생성
- 플랫폼과 분리하여 제품 개발에만 집중 가능



```
1 import React from 'react';
2 import { <%= fragmentName %> } from './__generated__<%= fragmentName %>';
3
4 type Props = Omit<%= fragmentName %>, '__typename';
5
6 function <%= componentName %>({ section }: Props) {
7   const { title } = section;
8
9   return <div>Time to get this component built!</div>;
10 }
11
12 export default <%= componentName %>;
13
```

```
1 import gql from 'apollo/client';
2
3 export default gql`
4   fragment <%= fragmentName %> on <%= sectionName %> {
5
6   }
7 `;
8
```

```
1 import React from 'react';
2 import createVariationProvider from ':react-component-variations/src/createV
3 import <%= componentName %> from '=<%= componentNameImport %>';
4 import manualMock from '=<%= manualMockImport %>';
5
6 export default createVariationProvider(() => ({
7   component: <%= componentName %>,
8   variations: [
9     {
10       title: 'default',
11       render: () => <<%= componentName %> {...manualMock} />,
12     },
13   ],
14 }));
15
```

GraphQL IDL

Thrift기반의 구조를 GraphQL 쿼리 문법에 친화적으로 변경

```

1 struct Screen {
2   1: optional string id
3   2: optional set<Layout> layouts
4 }
5
6 struct SectionContainer {
7   1: optional string id
8   2: optional SectionComponentType sectionComponentType
9   3: optional AvailableSection section
10 }
11
12 union AvailableSection {
13   1: optional Title
14   2: optional Banner
15 }
16
17 struct Banner {
18   1: optional Icon icon
19   2: optional string title
20   3: optional string subtitle
21   4: optional string ctaCopy
22   5: optional string ctaUrl
23 }
24
25 struct Title {
26   1: optional string title
27   2: optional string subtitle
28 }

```

```

1 query CheckoutPage($id: ID!) {
2   presentation {
3     checkout(productId: $id)
4     @serverMock(mockIdentifier: "loggedIn") {
5       sections {
6         sections {
7           ... on TitleSection {
8             title
9           }
10          ... on BannerSection {
11            title
12            subtitle
13            ctaCopy
14            ctaUrl
15          }
16        }
17      }
18    }
19  }
20 }
21

```

```

WishlistIndexQueryResponse response =
    ViaductClient.execute(
        WishlistsIndexQuery.builder()
            .limit(10)
            .build()
    );

response.getViewer().getFirstName();

```

```

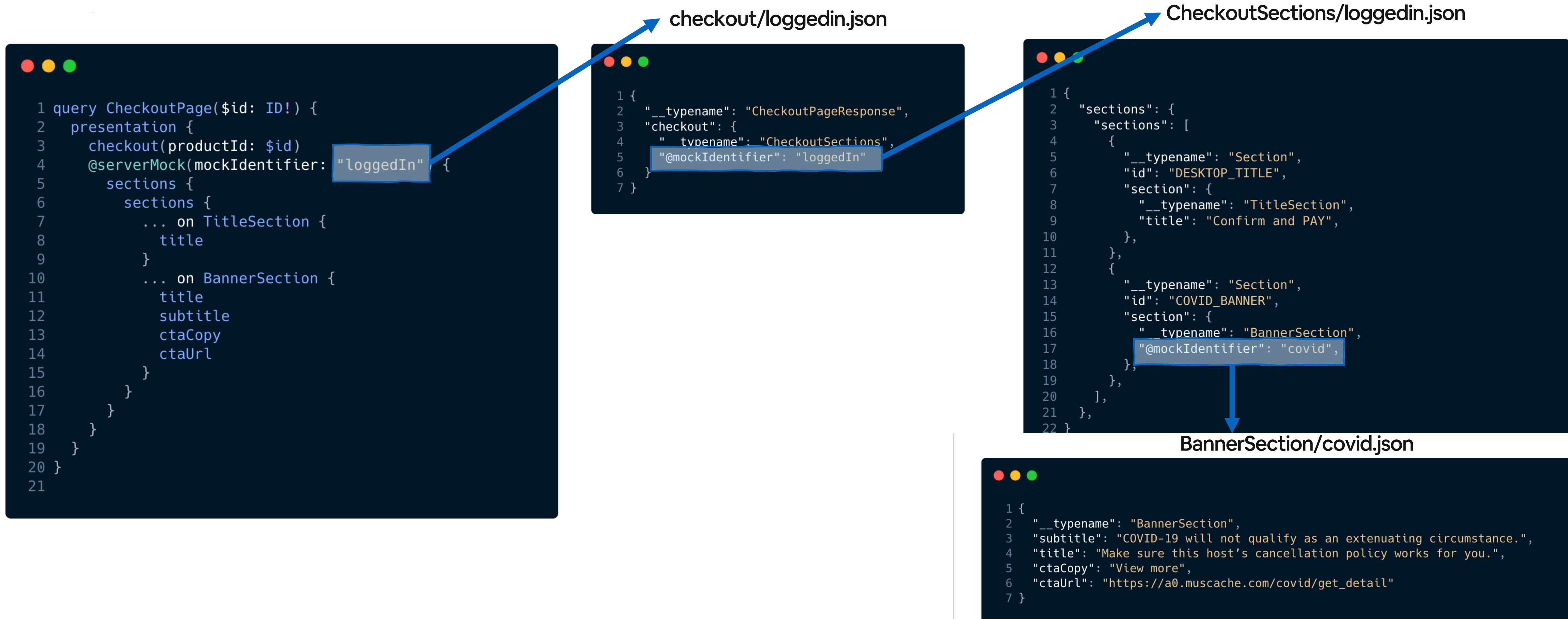
class ViewerFullNameProvider : CoroutineDerivedFieldProvider<String> {
    override fun getFragment(args: FragmentArguments): FragmentDefinition {
        // language=GraphQL
        return fragment(
            """
            fragment ViewerFullNameProvider on Viewer {
                firstName
                lastName
            }
            """.trimIndent()
        )
    }

    override suspend fun resolve(context: DerivedFieldProviderContext): String {
        val obj = context.getSourceObject()
        return "${obj["firstName"] ?: ""} ${obj["lastName"] ?: ""}".trim()
    }
}

```

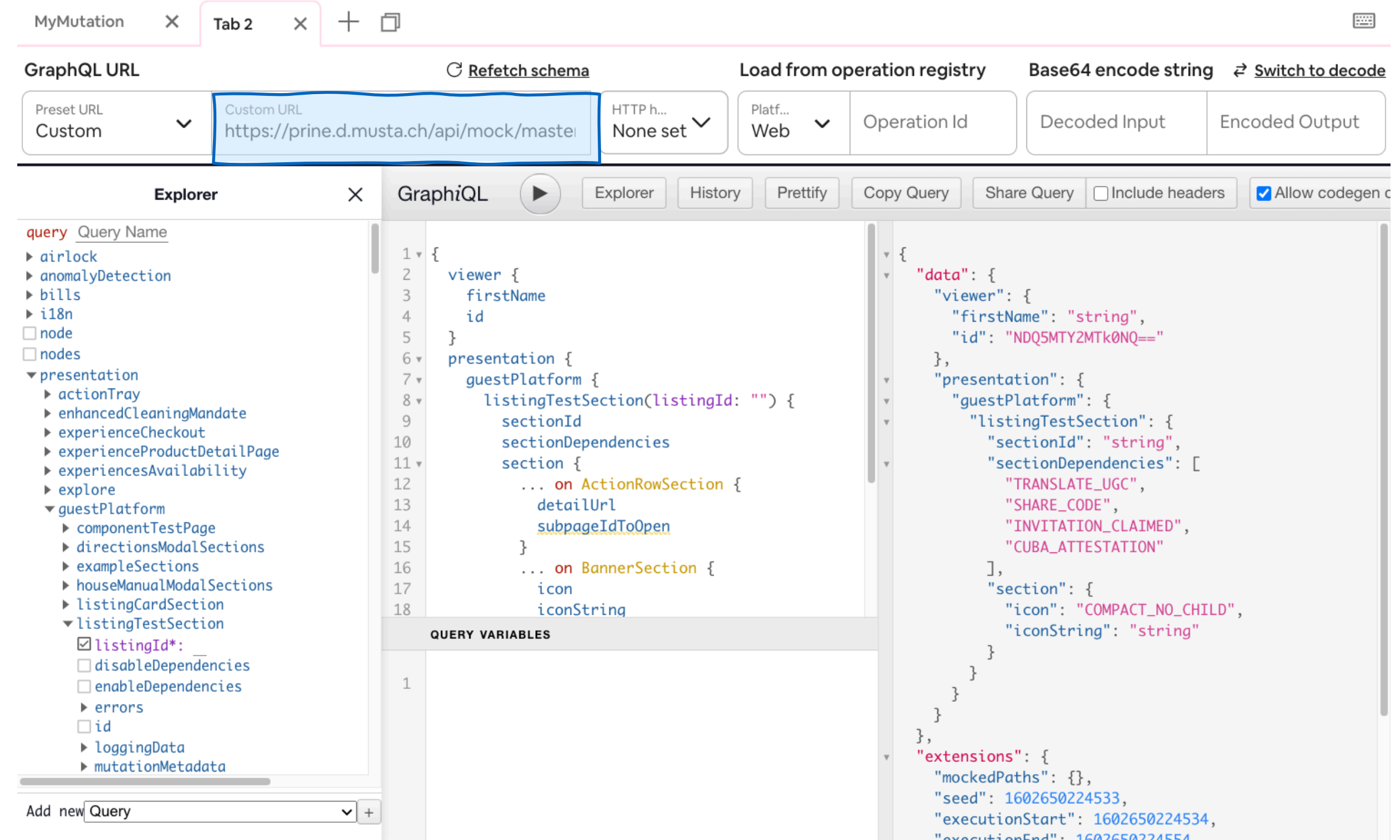
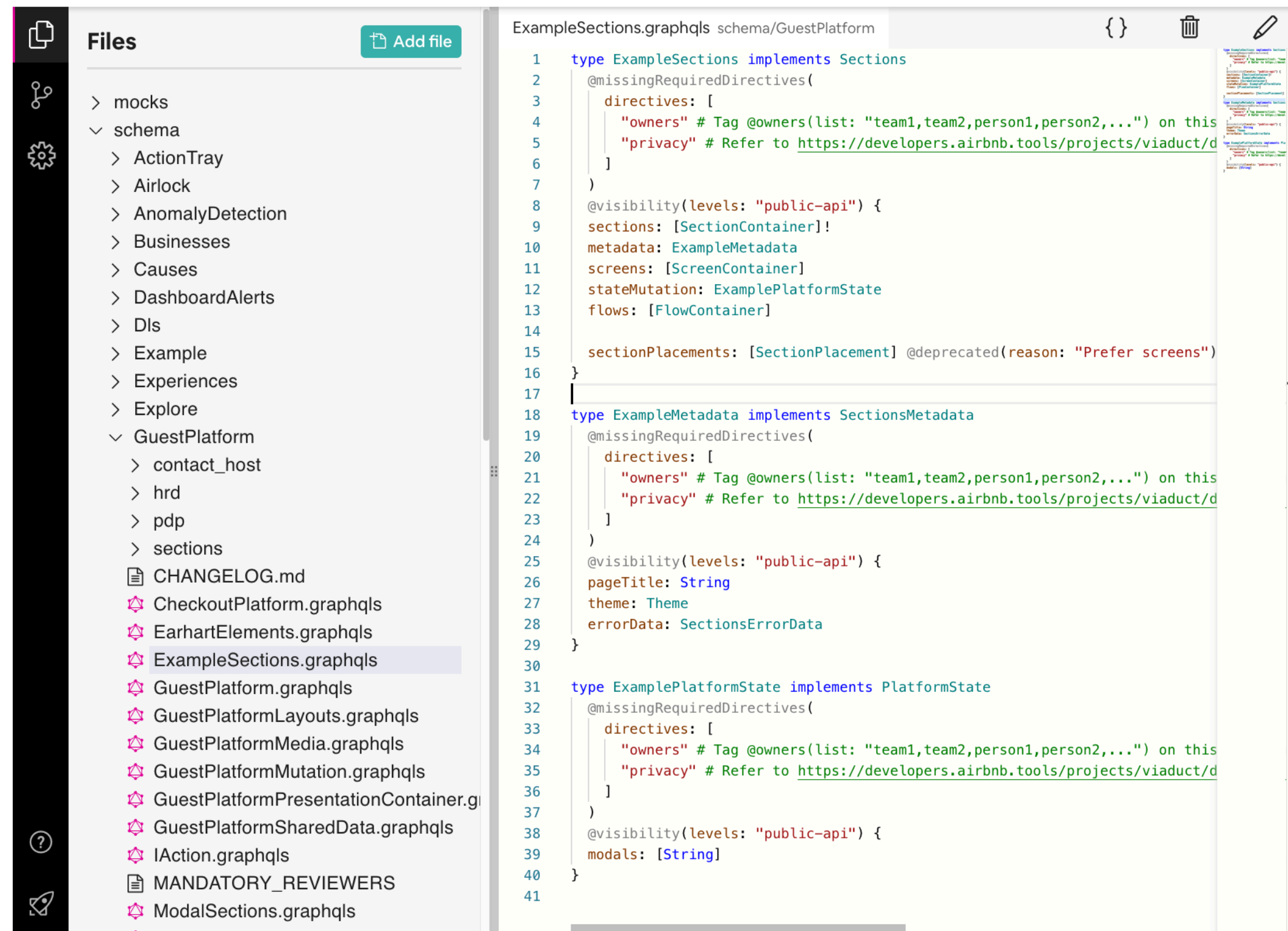
Data Mocking

기본 결과값으로 mockIdentifier에 명시된 json파일

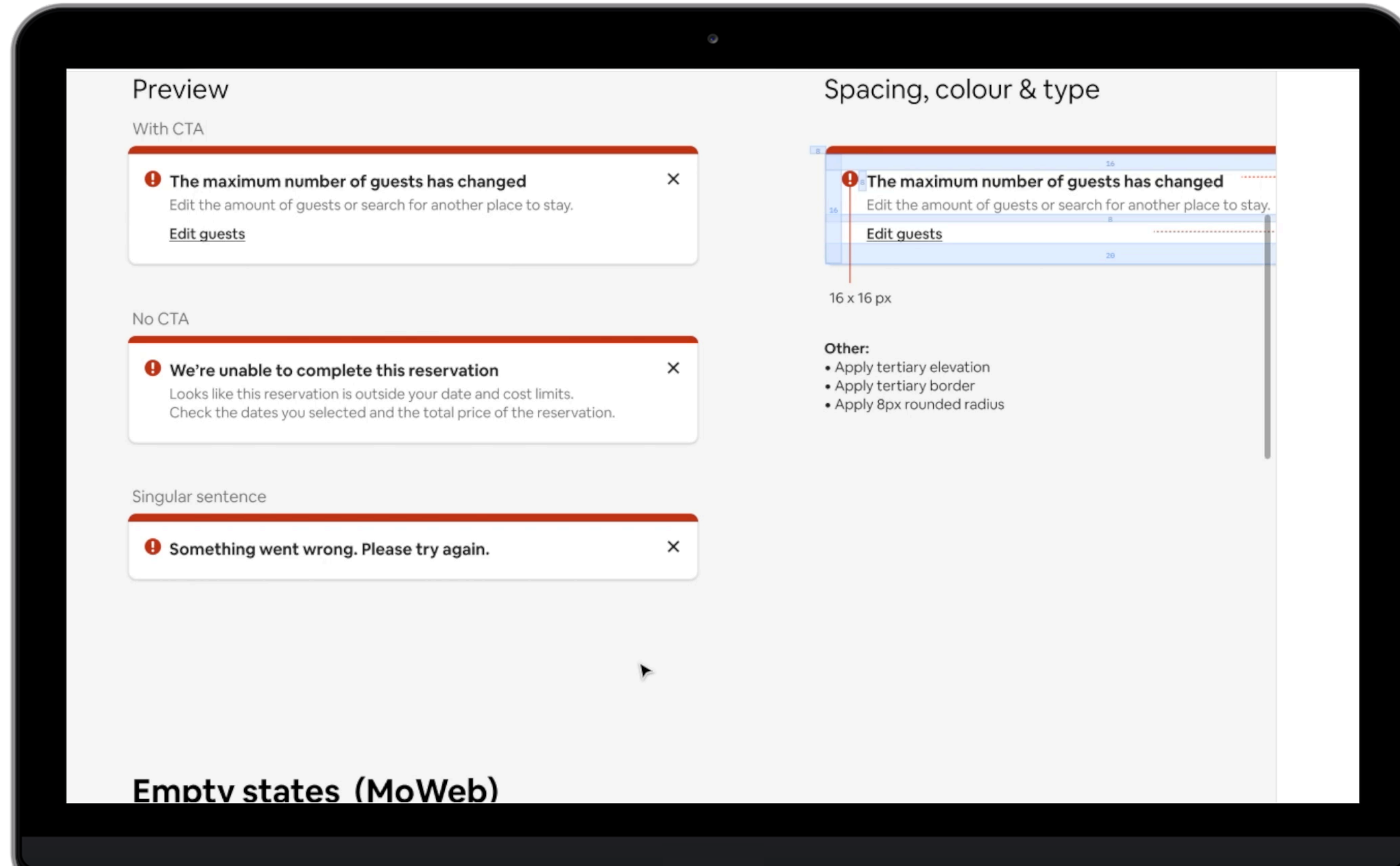


개발 중인 툴들

N DEVVIEW
2020



개발 과정 시연



긍정적인 효과들

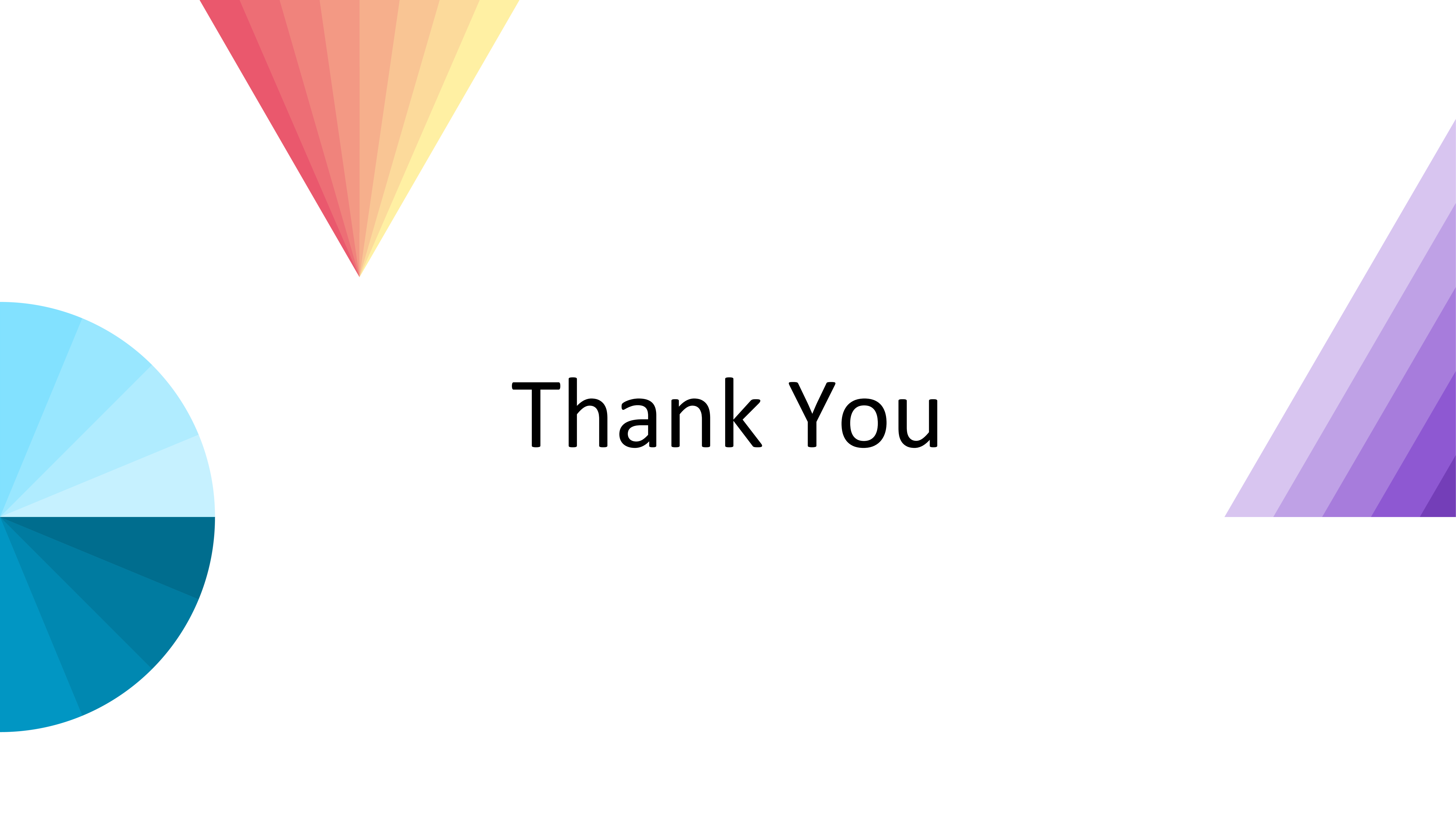
- 생산성
- 재활용
- 손쉬운 A/B 테스트
- 개발 비용 최소화 프로토타입

고민해 볼 요소들

- 모두에게 적합하진 않음
- 초기 학습 장벽
- 너무 많은 설정 파일들
- 개발 환경 투자 비용



Q & A



Thank You