Build "Automation Test as a Service" for large-scale e-commerce website in eBay

Bingsheng Ru (Robin Ru)

Engineering Productivity Tech Leader
eBay China Technology Center of Excellence

SPEAKER

INTRODUCTION

Bingsheng Ru （ Robin ）

- eBay – **Engineering Productivity Tech Leader**
- Hewlett-Packard – **Test Architect, Test Expert**
- Alcatel-Lucent – **Senior Test Leader**
- Cisco – **Senior Quality Engineer**

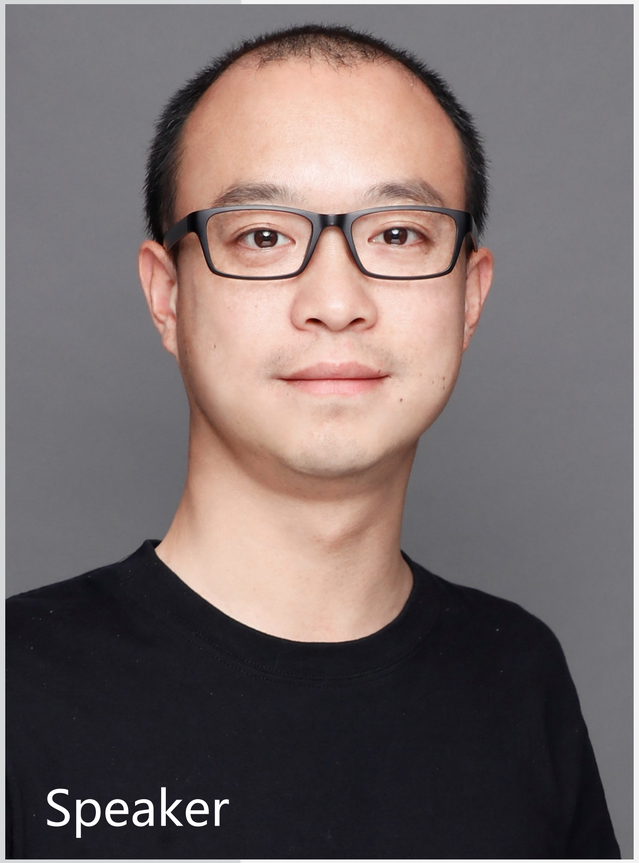
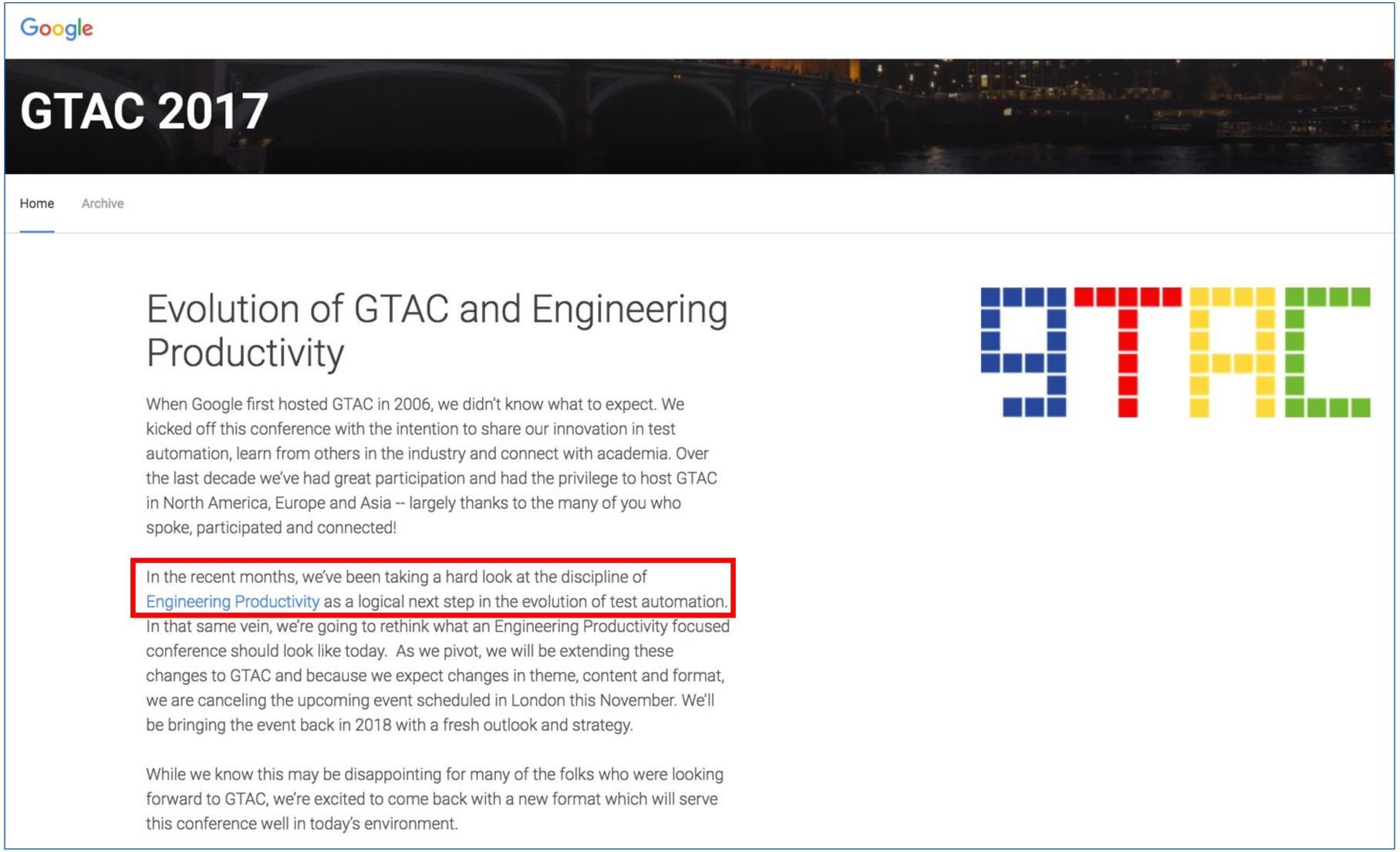

Speaker

TABLE OF
CONTENTS

- From Quality Engineering to Engineering Productivity
- Automation Test as a Service
  - Test Data Service
  - Test Execution Service
  - Test Bed Service
  - Global Registry Service

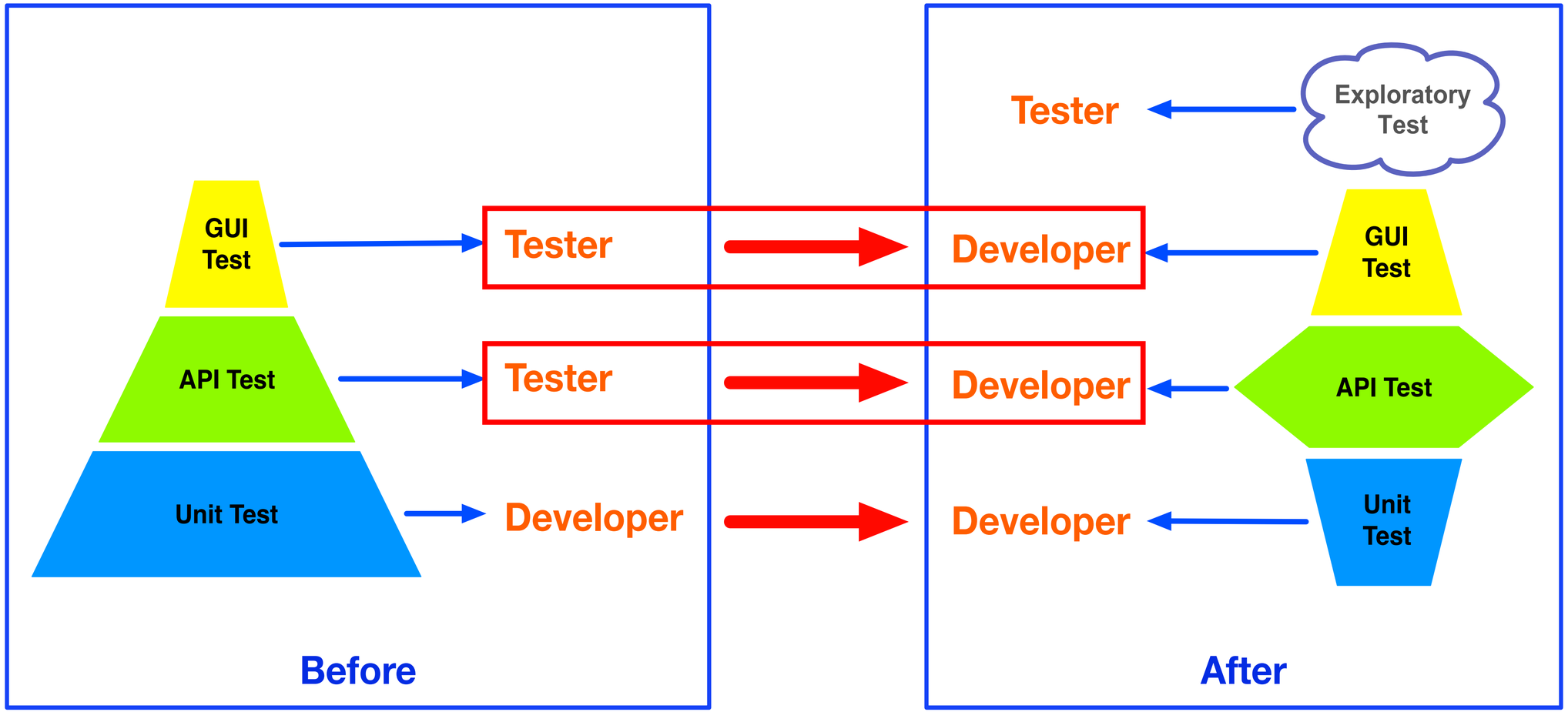
From Quality Engineering to Engineering Productivity

Let's start from Google



https://testing.googleblog.com/2016/03/from-qa-to-engineering-productivity.html

From Quality Engineering to Engineering Productivity

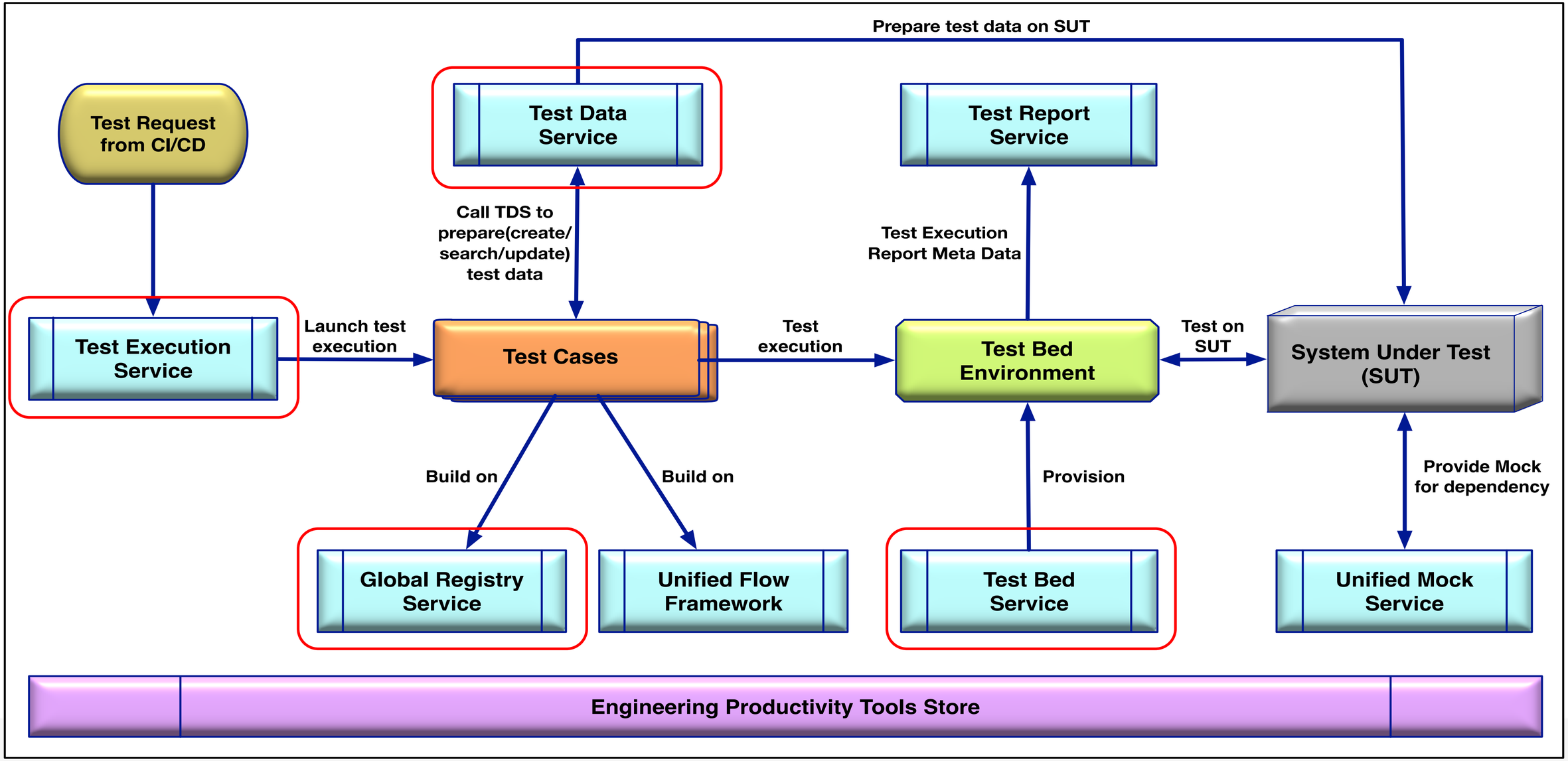


From Quality Engineering to Engineering Productivity

Strong Test Harness
=
Automation Test as a Service

Automation Test as a Service

Automation Test as a Service



Test Data Service

Test Data Service

Test Data Challenge

- Complexity for test data combination
- Time consumption for On-the-fly test data
- Dirty data for Out-of-box test data
- Time consumption for performance test data
- For Microservice arch, not enough knowledge for creating test data due to cross domain
- For Microservice arch, test data highly depend on test environment

Test Data Service

Test Data Preparation 1.0 Phase – Java-based Test Data Utility

```
145     public static User createUser(String userName, String password, Contact primaryContact,
146         Contact paymentContact, boolean enable2FA) throws Exception {
```

```
199     public static String createListing(String eventId, boolean isGA, boolean isParking_Pass,
200         List<Ticket> tickets, Delivery delivery,
201         Price pricePerProduct, Price faceValue, String quantity, ListingSplit listingSplit,
202         String inhandDate, String saleEndDate,
203         String username, String password) {
```

Test Data Service

Test Data Preparation 2.0 Phase – Java-based Build Pattern

```
createUserImpl(A, B, C, D, E, ...);

createDefaultEvent()
{
    init A,B,C,D,E,...
    createUserImpl(A, B, C, D, E, ...);
}

createSellerUser(A)
{
    init B,C,D,E,...
    createUserImpl(A, B, C, D, E, ...);
}

createBuyerUser(B, C)
{
    init A,D,E,...
    createUserImpl(A, B, C, D, E, ...);
}
```

```
UserBuilder.build();
UserBuilder.withA(A).build();
UserBuilder.withA(B).withA(C)
            .withBuildStrategy(BuildStrategy.CREATE_ONLY)
            .build();

UserBuilder.withRecipe(XXX).build();
```

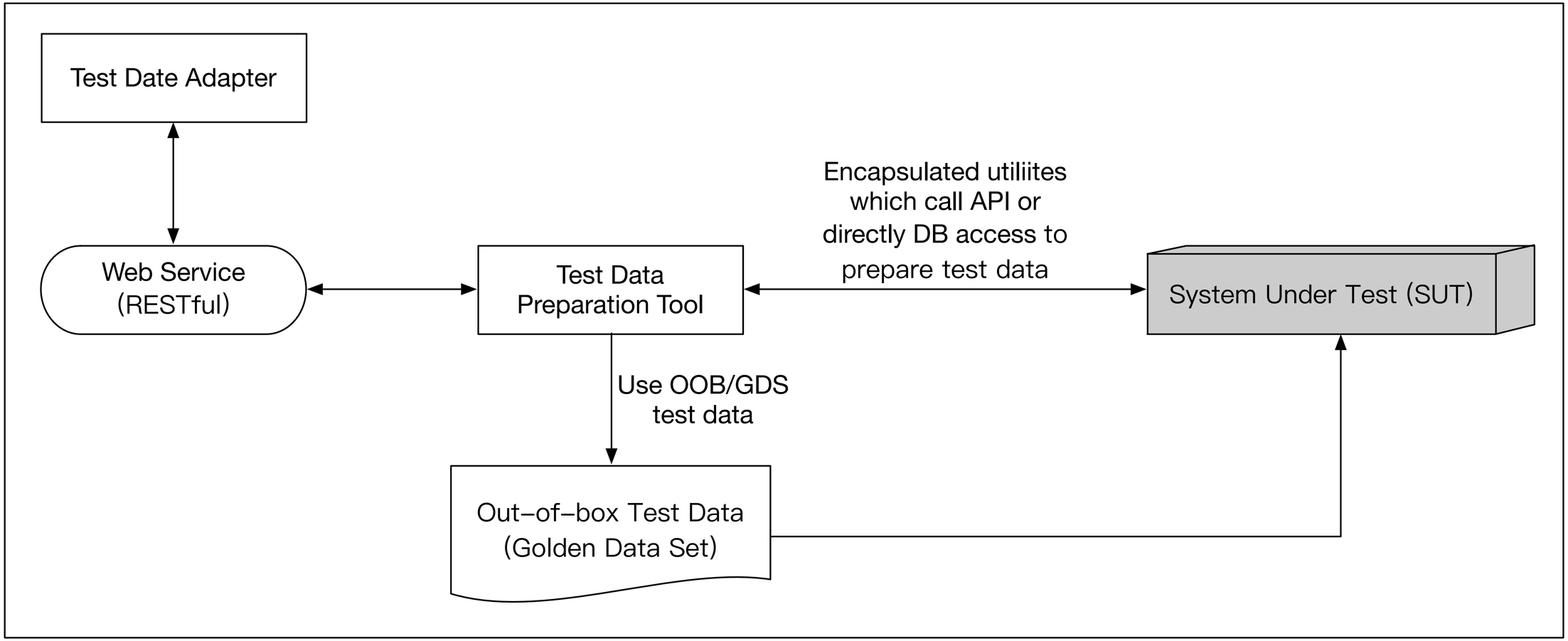
```
BuildStrategy.CREATE_ONLY
BuildStrategy.SEARCH_ONLY
BuildStrategy.SMART
BuildStrategy.GOLDENT_DATA_ONLY
```

```
User seller = GlobalUserBuilder.newBuilder().withSellerType(SellerType.valueOf(sellerType))
    .withCreditCard().withCountry(CountryUtil.getCountry(sellerCountryCode)).build();
```

Test Data Service

Test Data Preparation 3.0 Phase – Test Data Service

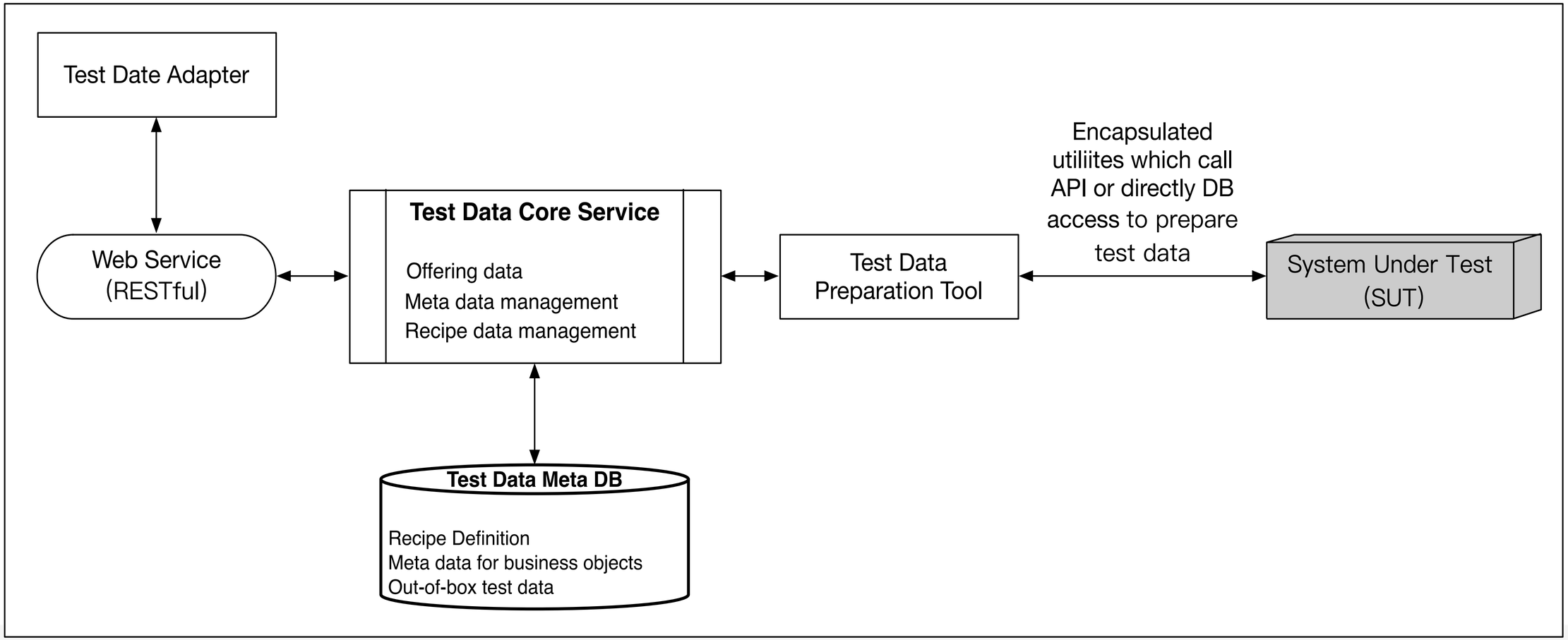
Test Data Service Rudiment



Test Data Service

Test Data Preparation 3.0 Phase – Test Data Service

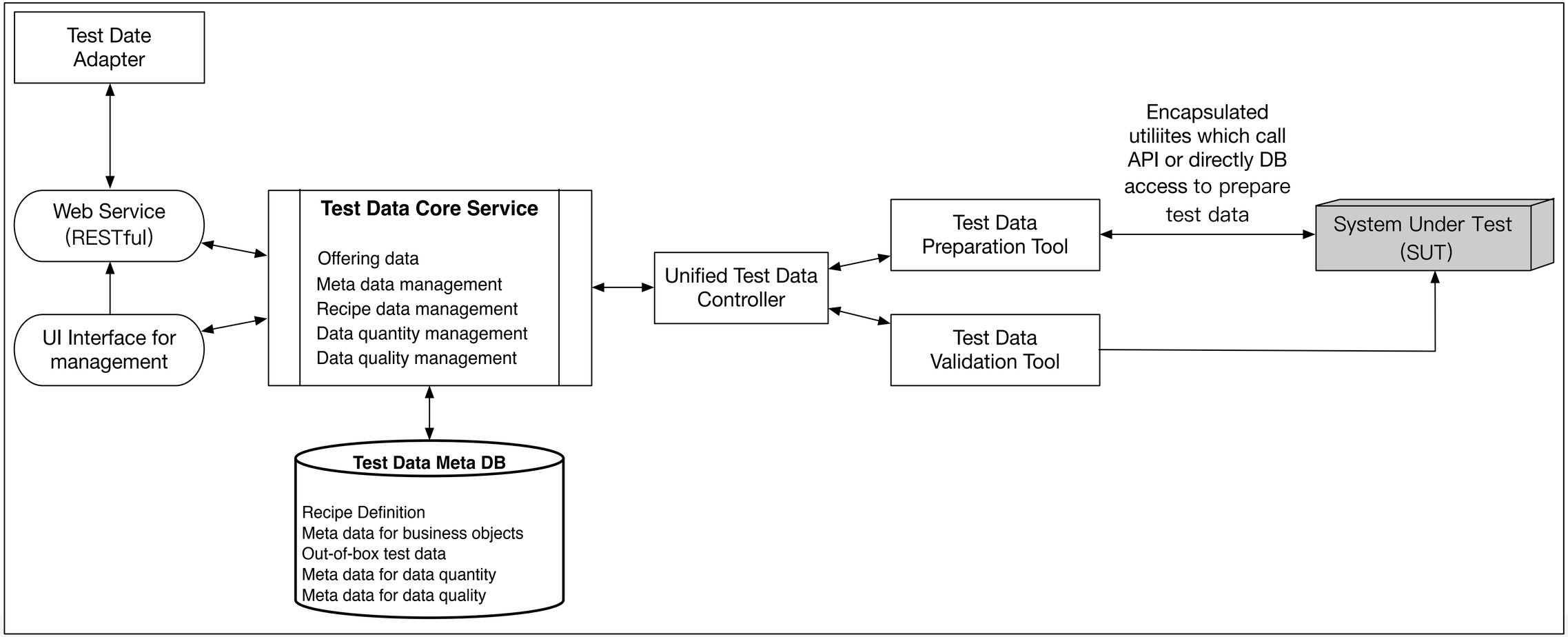
Introduce Test Data Core Service and Internal Database



Test Data Service

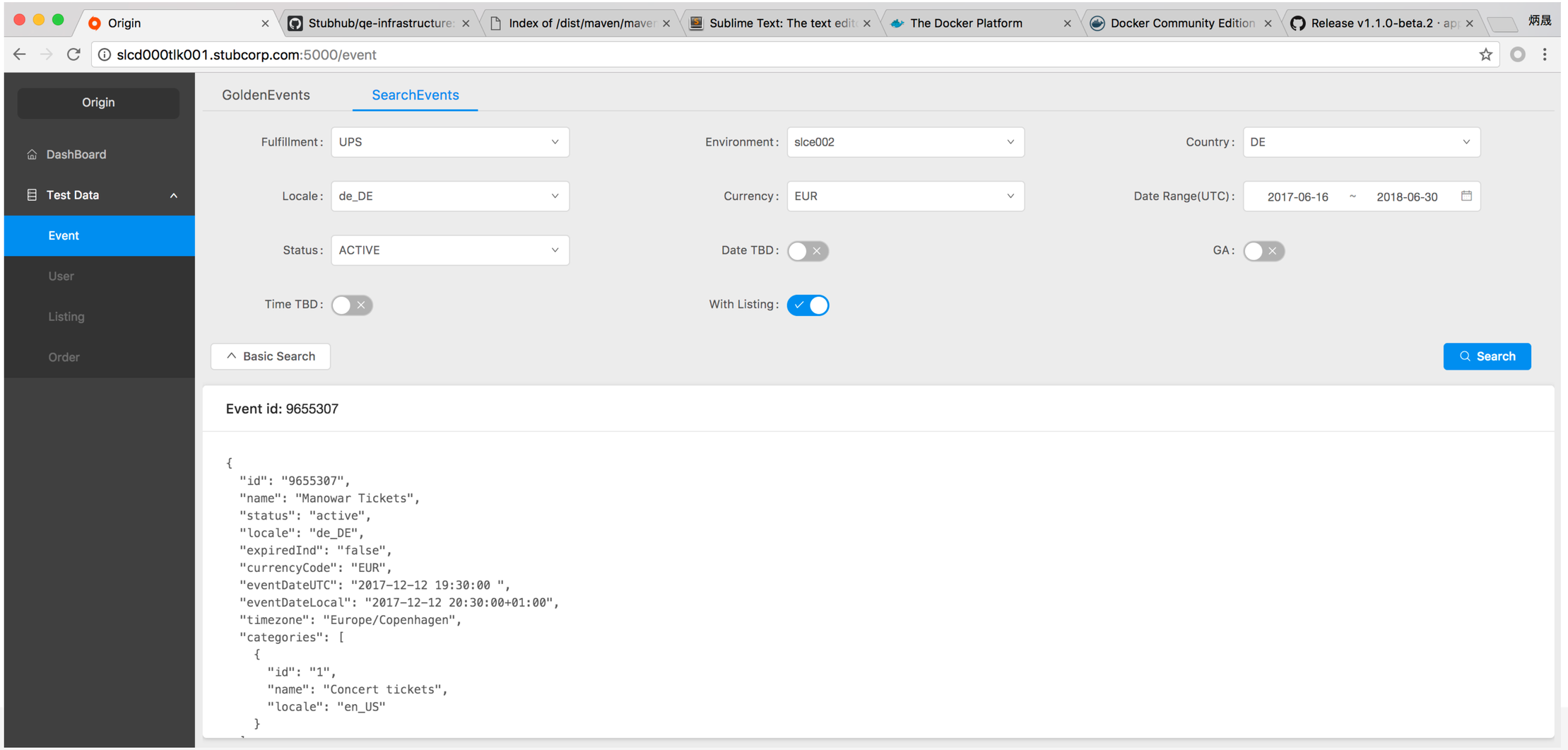
Test Data Preparation 3.0 Phase – Test Data Service

Introduce Data Quantity / Quality management



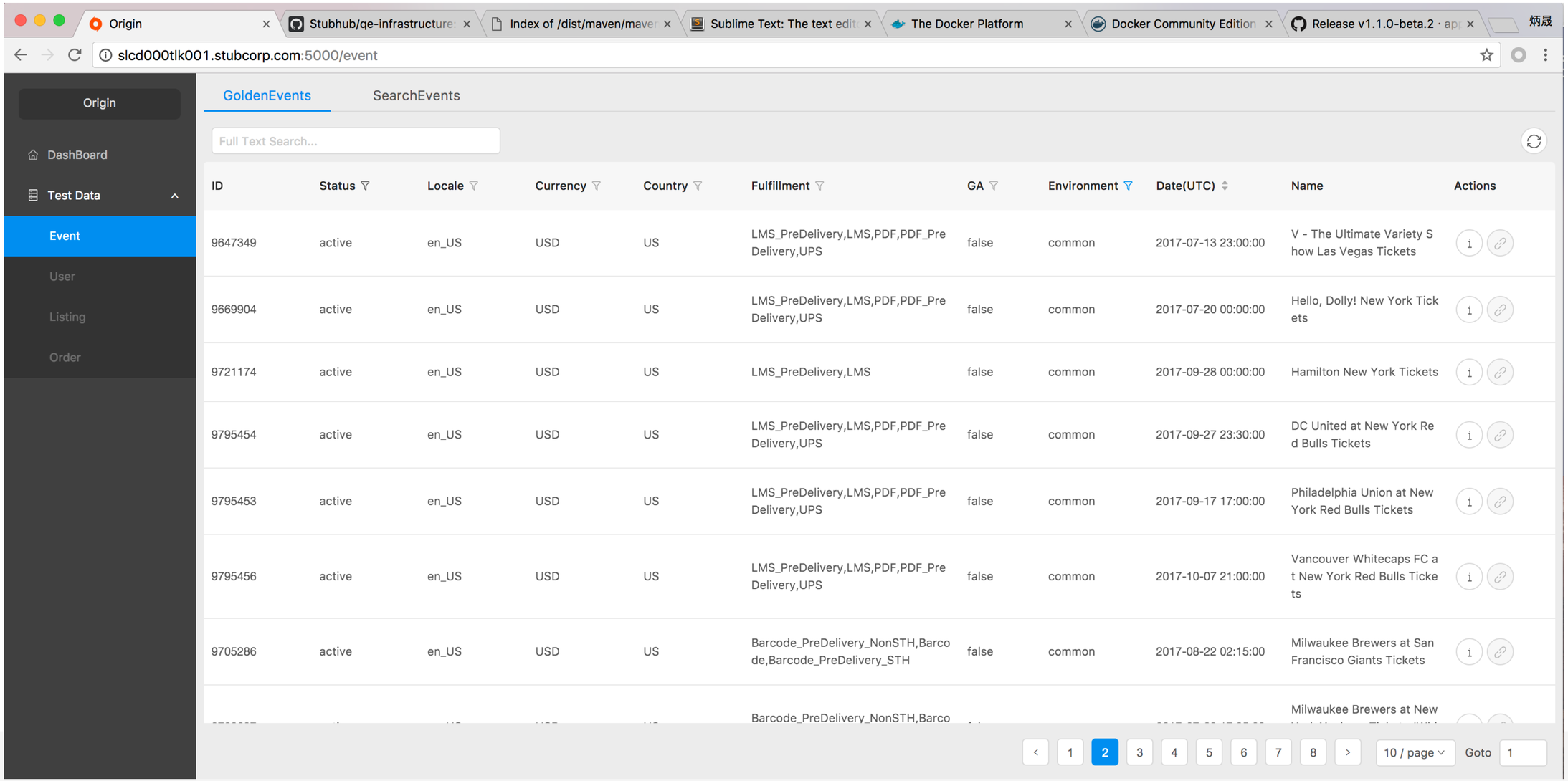
Test Data Service

Test Data Service GUI Screenshot



Test Data Service

Test Data Service GUI Screenshot



Test Execution Service
and
Test Bed Service

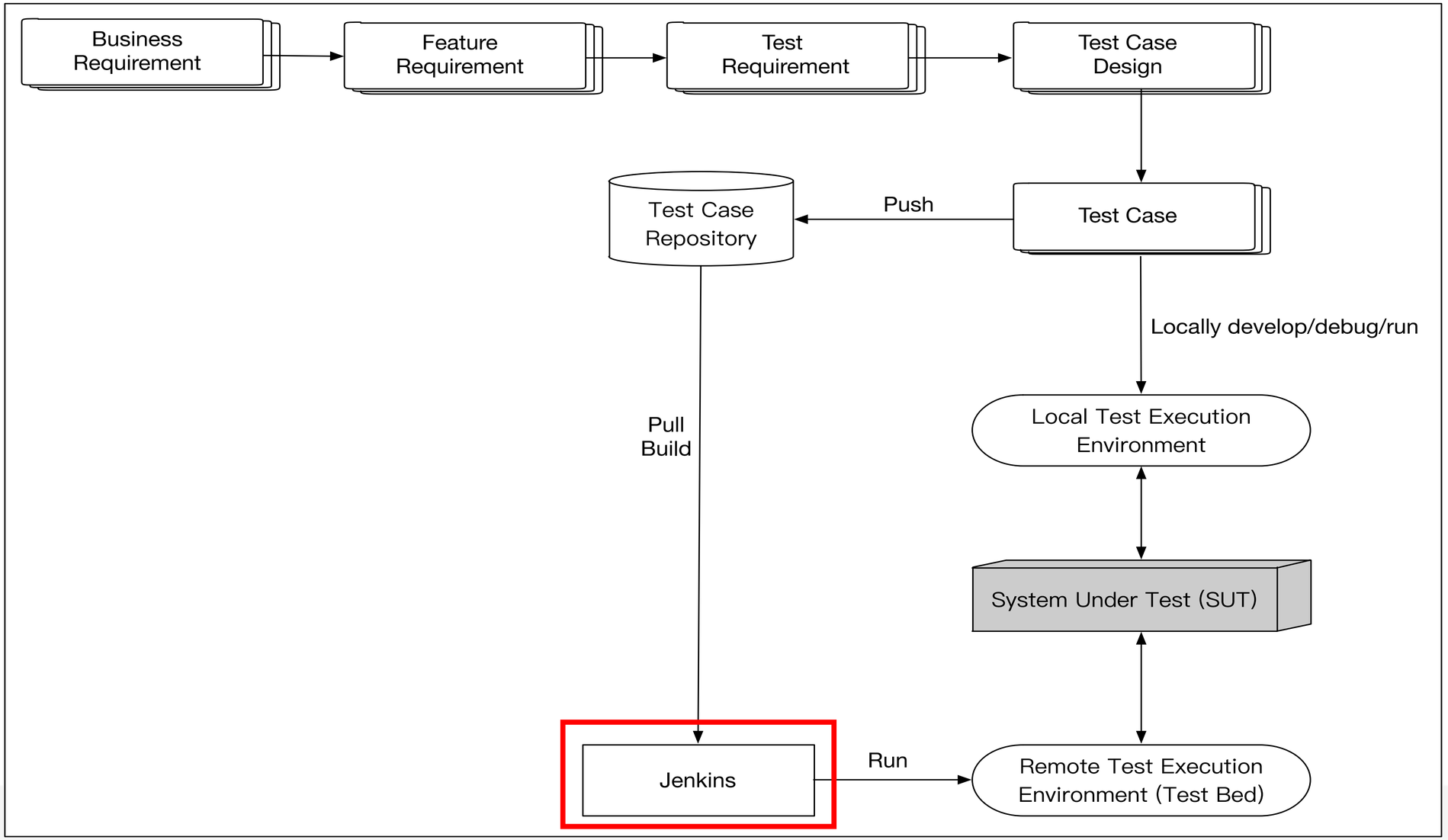
Test Execution Service and Test Bed Service

What is the ideal Test Execution Platform

- Transparency to user(DEV, CI/CD pipeline)
- Ease maintenance to Maintainer
- Scalability to test case execution

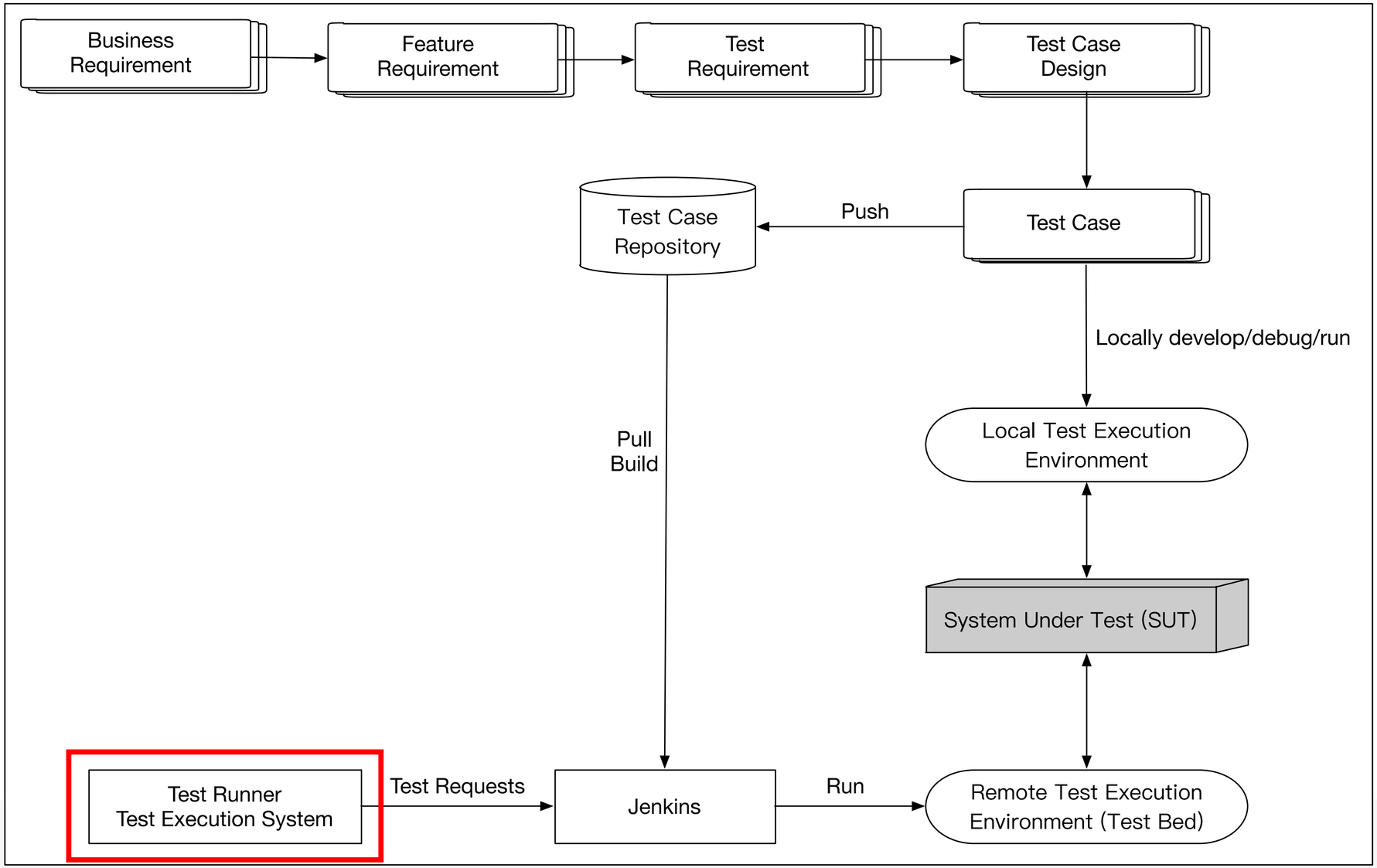
Test Execution Service and Test Bed Service

Use Jenkins to trigger test execution



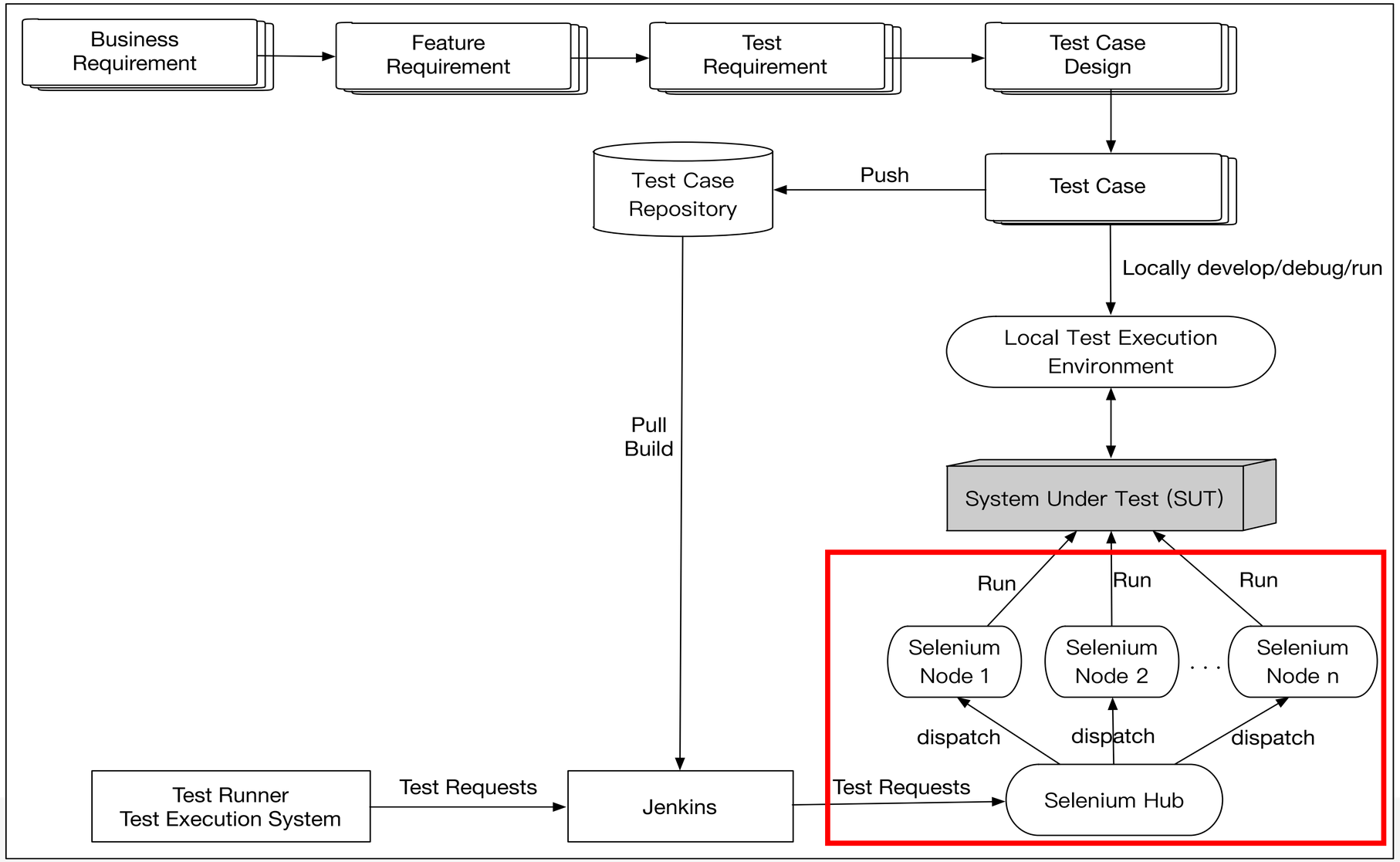
Test Execution Service and Test Bed Service

Introduce Test Execution System



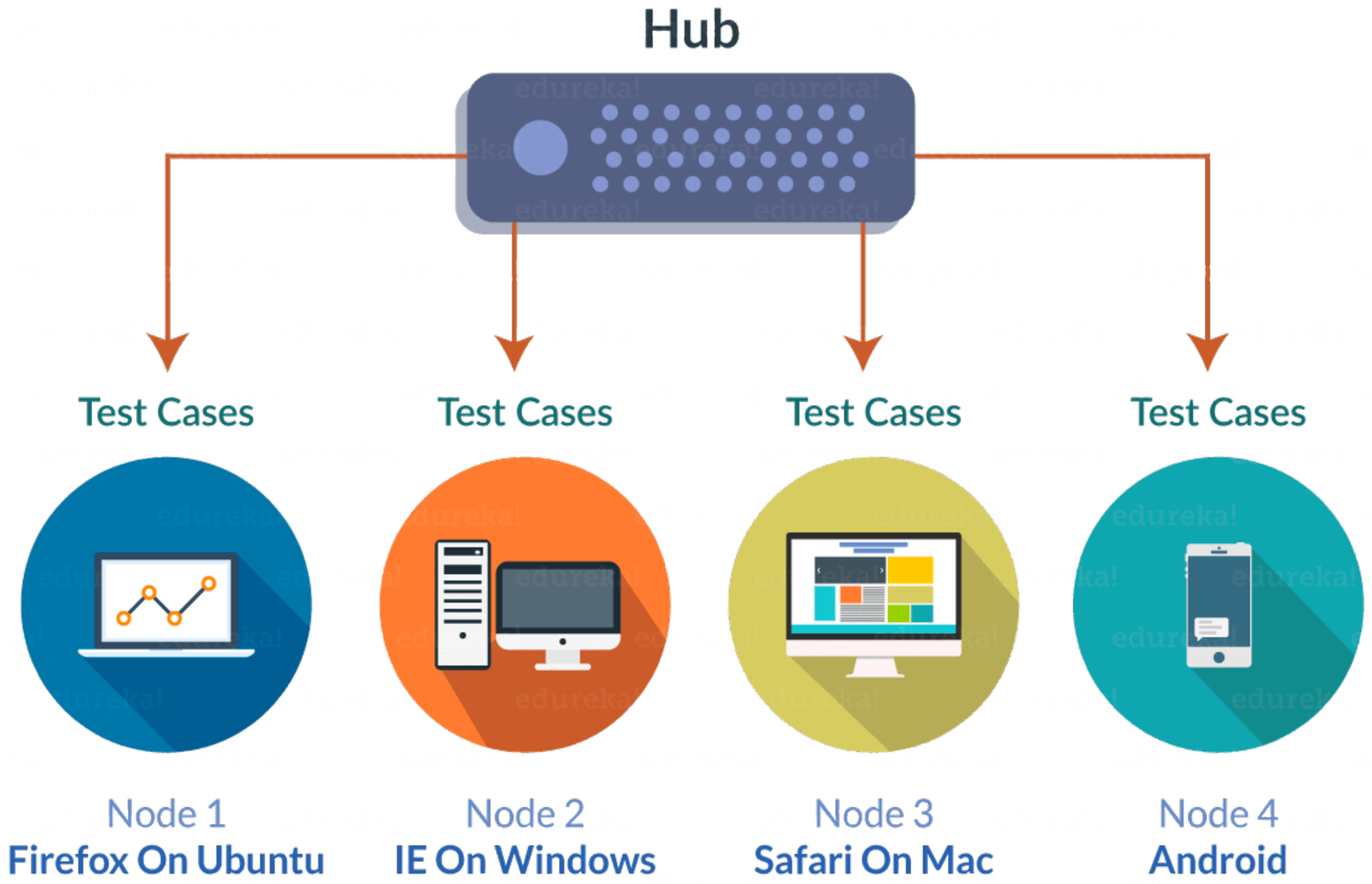
Test Execution Service and Test Bed Service

Introduce Selenium Grid



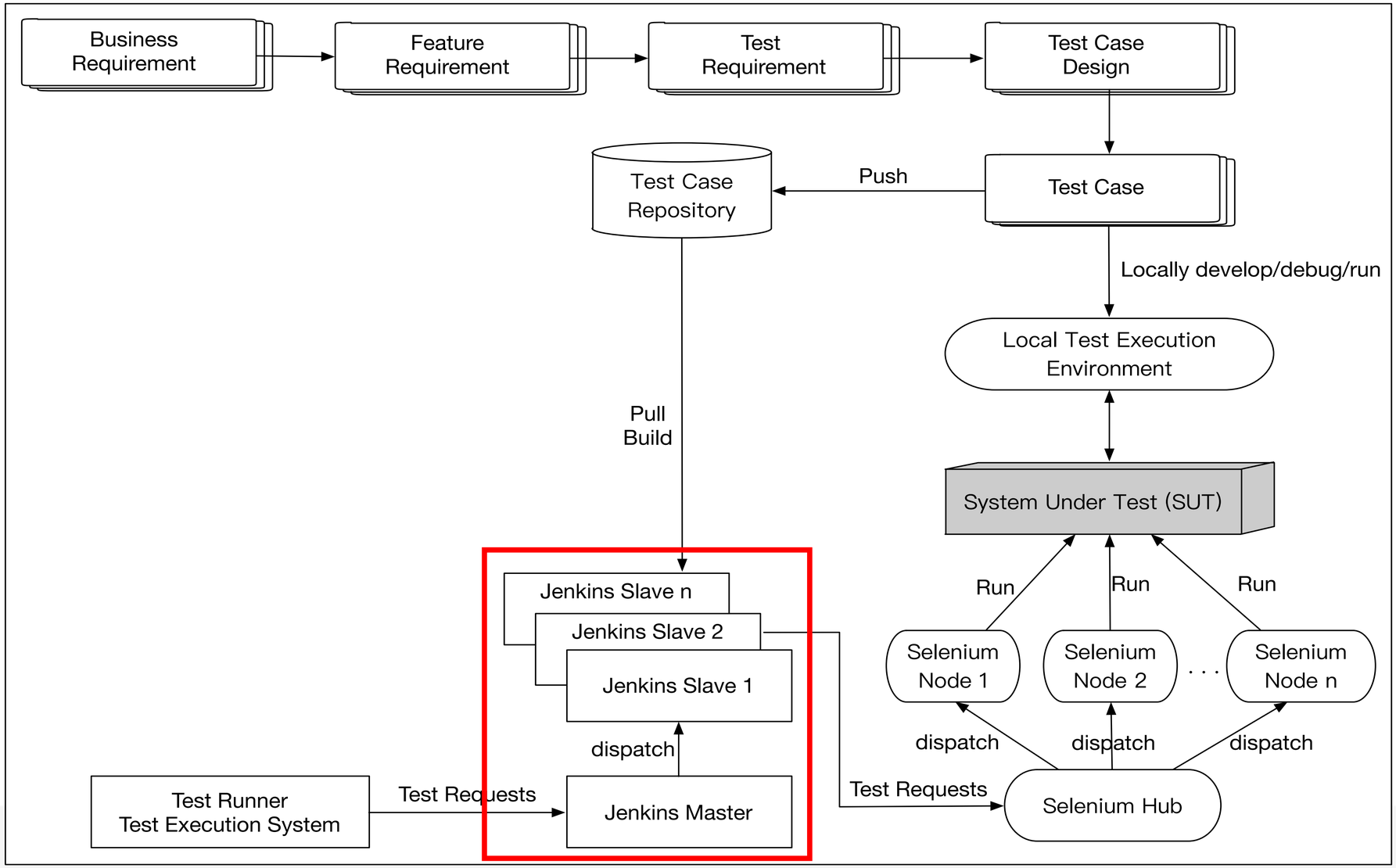
Test Execution Service and Test Bed Service

Selenium Grid



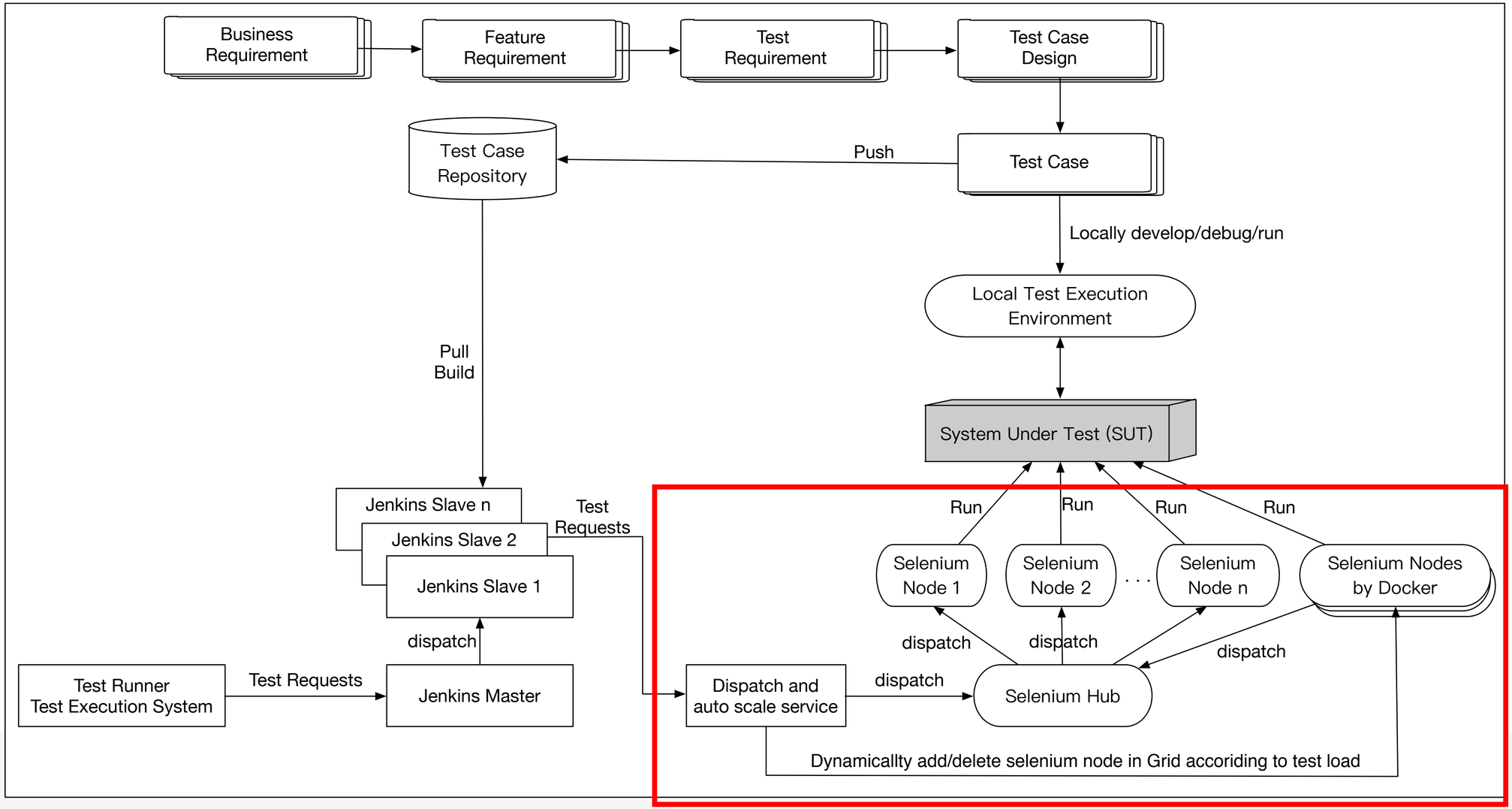
Test Execution Service and Test Bed Service

Use Jenkins cluster to improve test case distribution capacity



Test Execution Service and Test Bed Service

Enable auto-scale capacity for Selenium Grid to dynamically meet test load by using Docker



Test Execution Service and Test Bed Service

Challenge in Mobile Test Platform

- Android Fragmentation
- Different Device Coverage
- Test Execution Time

Test Execution Service and Test Bed Service

Build Mobile Test Execution Platform in Private Cloud

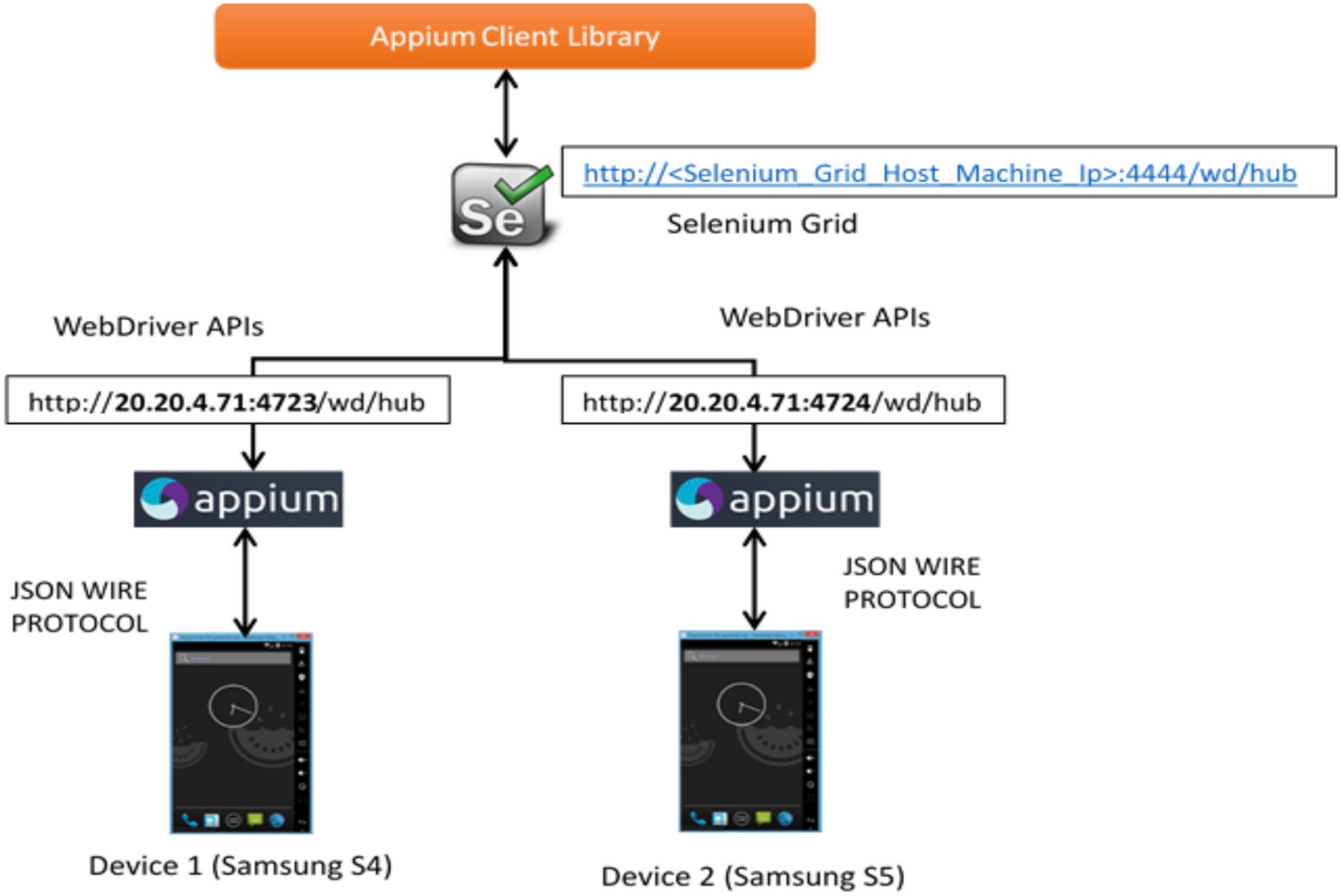
appium

Automation for Apps

Appium is an open source test automation framework for use with native, hybrid and mobile web apps.
It drives iOS, Android, and Windows apps using the WebDriver protocol.

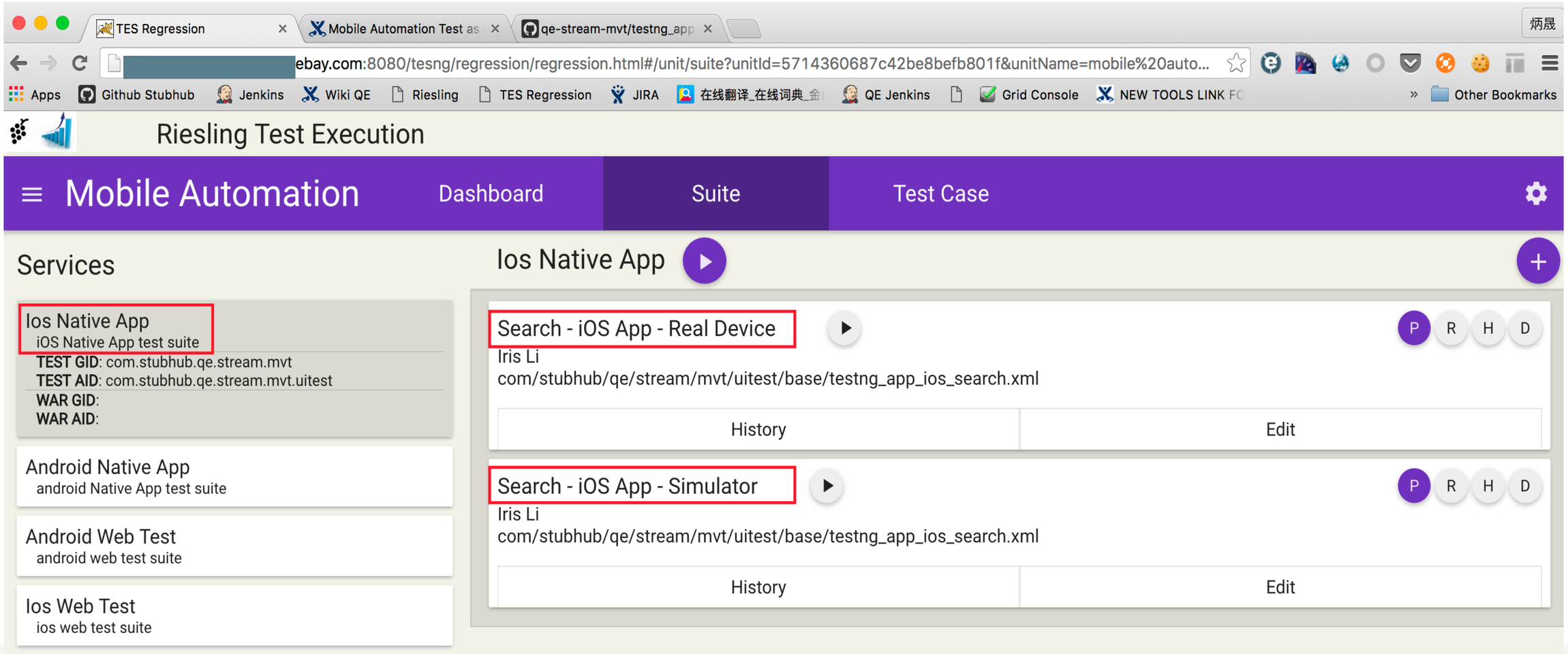
Test Execution Service and Test Bed Service

Base on Appium, build Mobile Test Execution Platform in Private Cloud



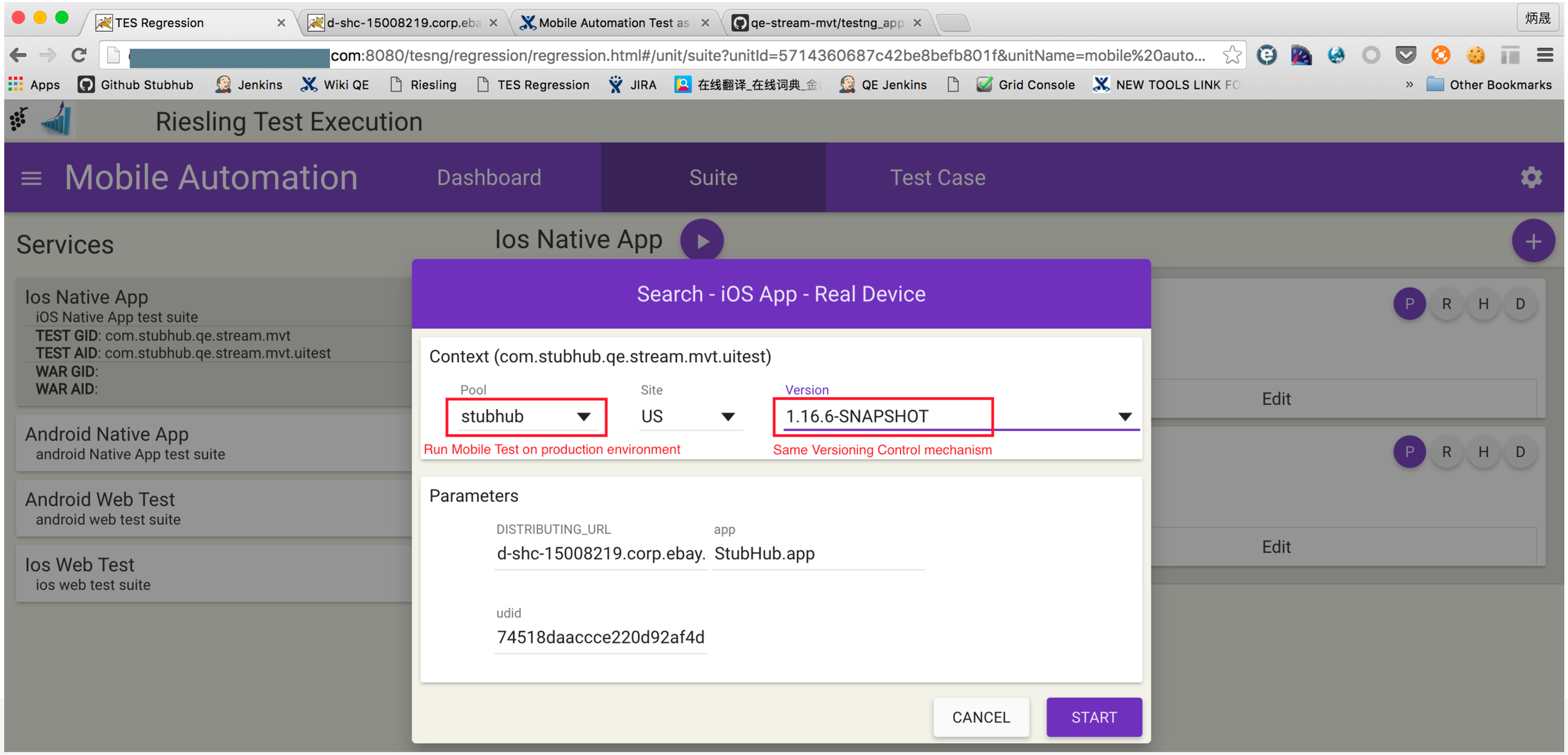
Test Execution Service and Test Bed Service

DEMO: Build Mobile Test Execution Platform in Private Cloud



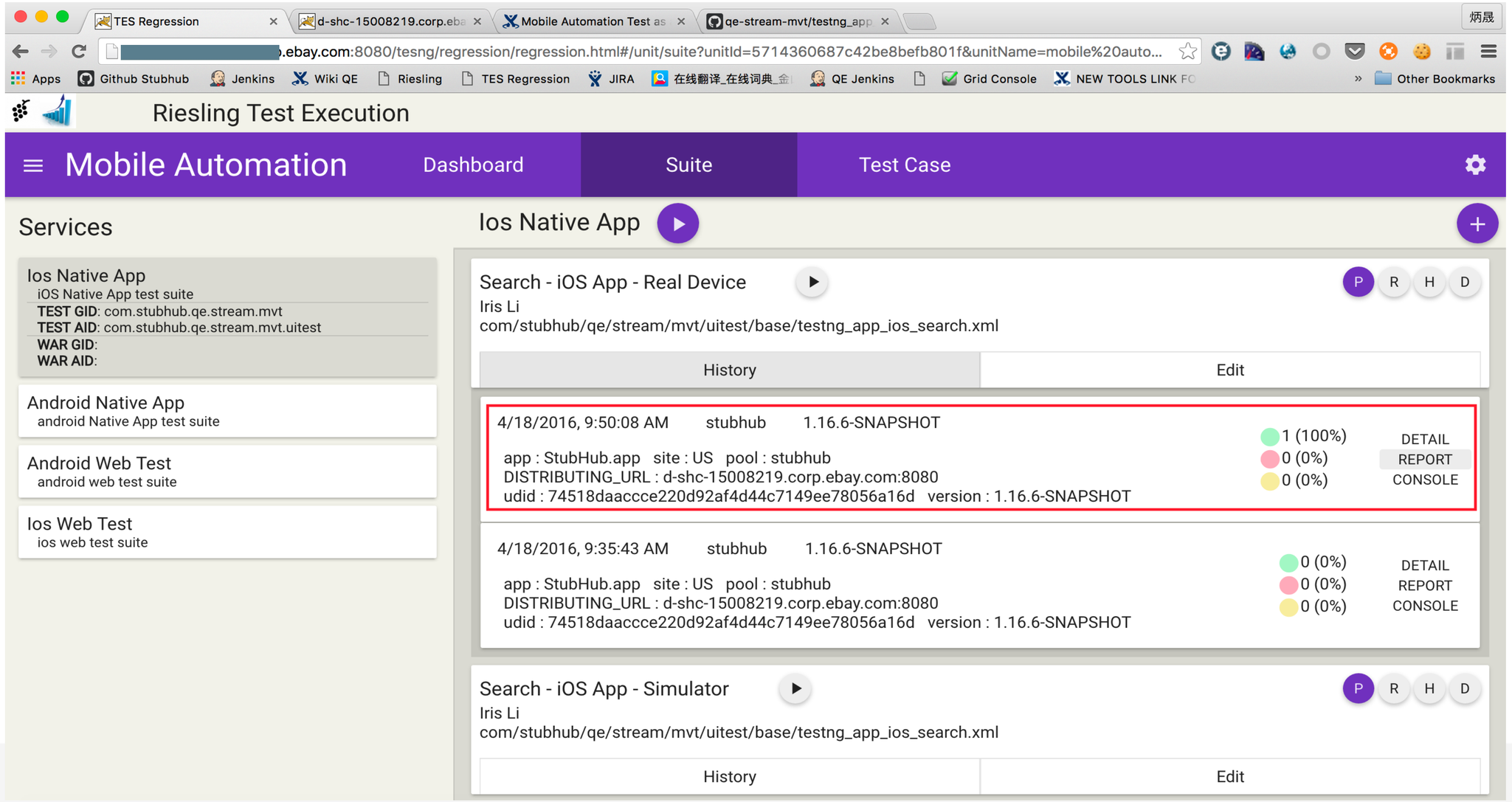
Test Execution Service and Test Bed Service

DEMO: Build Mobile Test Execution Platform in Private Cloud



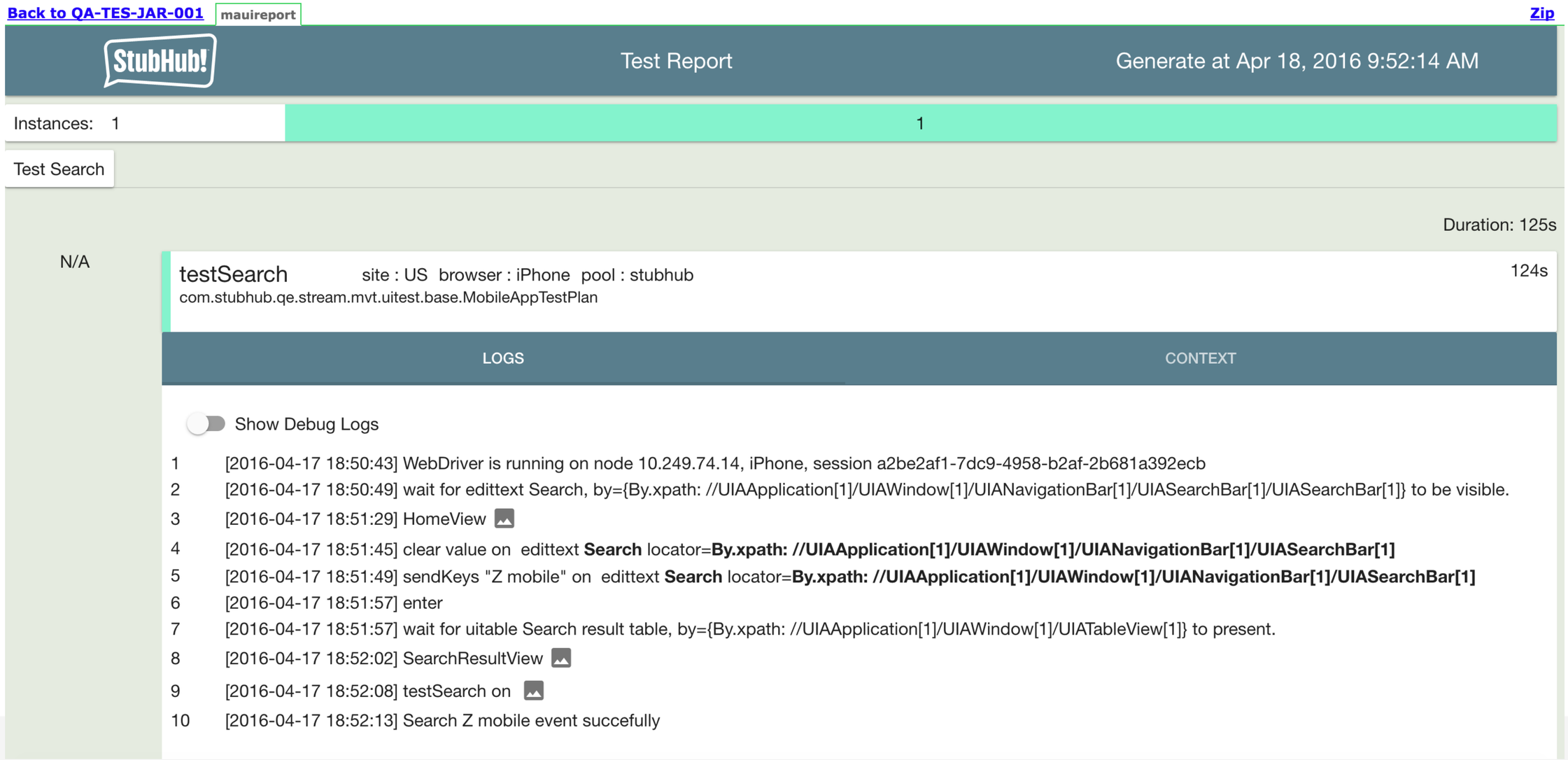
Test Execution Service and Test Bed Service

DEMO: Build Mobile Test Execution Platform in Private Cloud



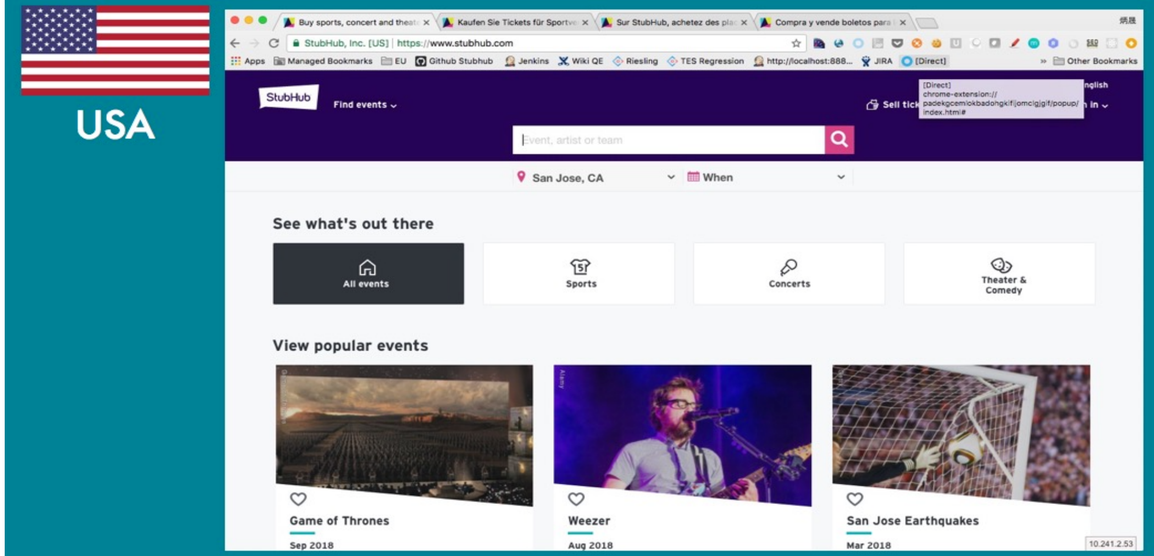
Test Execution Service and Test Bed Service

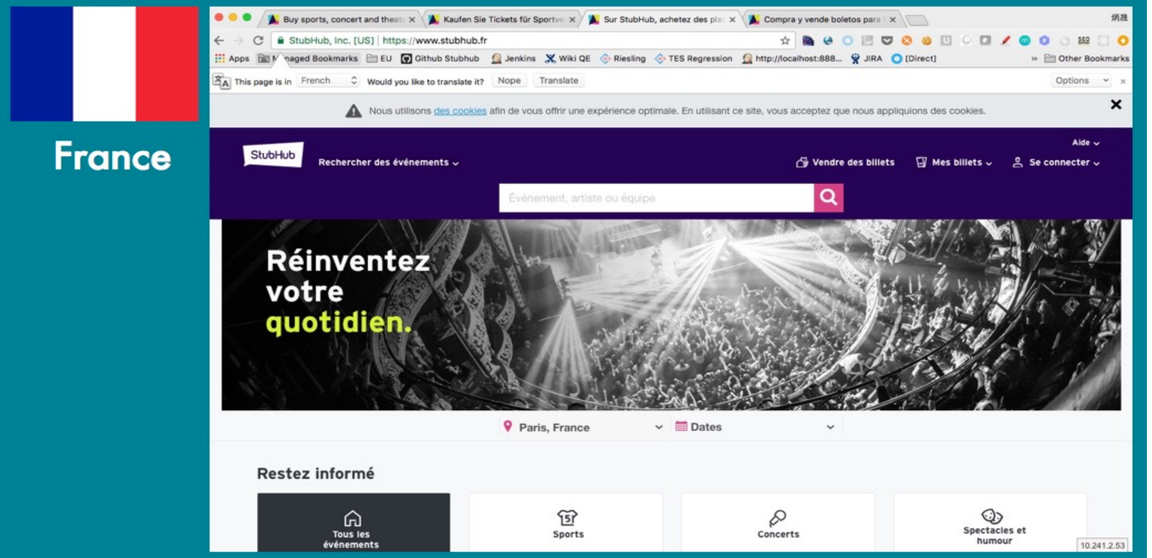
DEMO: Build Mobile Test Execution Platform in Private Cloud

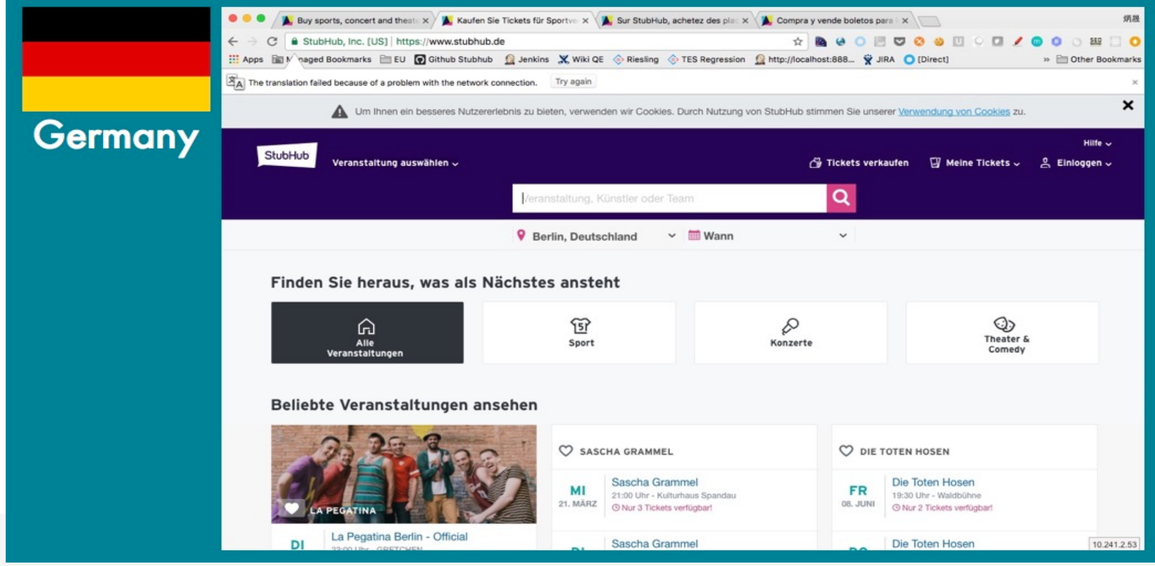


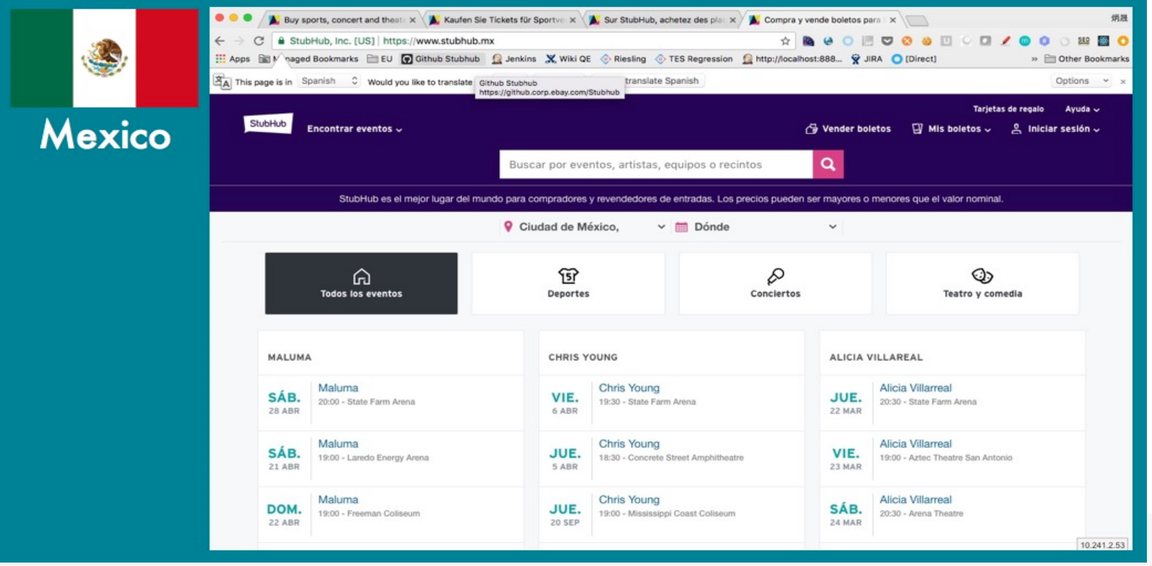
Global Registry Service

Global Registry Service









Global Registry Service

Before

```
public static String getCurrencyCode() {
    String currencyCode = "USD";
    if (Environment.isDESite() || Environment.isFRSite()) {
        currencyCode = "EUR";
    } else if (Environment.isUKSite()) {
        currencyCode = "GBP";
    } else if (Environment.isUSSite() || Environment.isMXSite()) {
        currencyCode = "USD";
    } else {
        throw new IllegalArgumentException("Site is not supported : " + Environment.getSite());
    }
    return currencyCode;
}
```

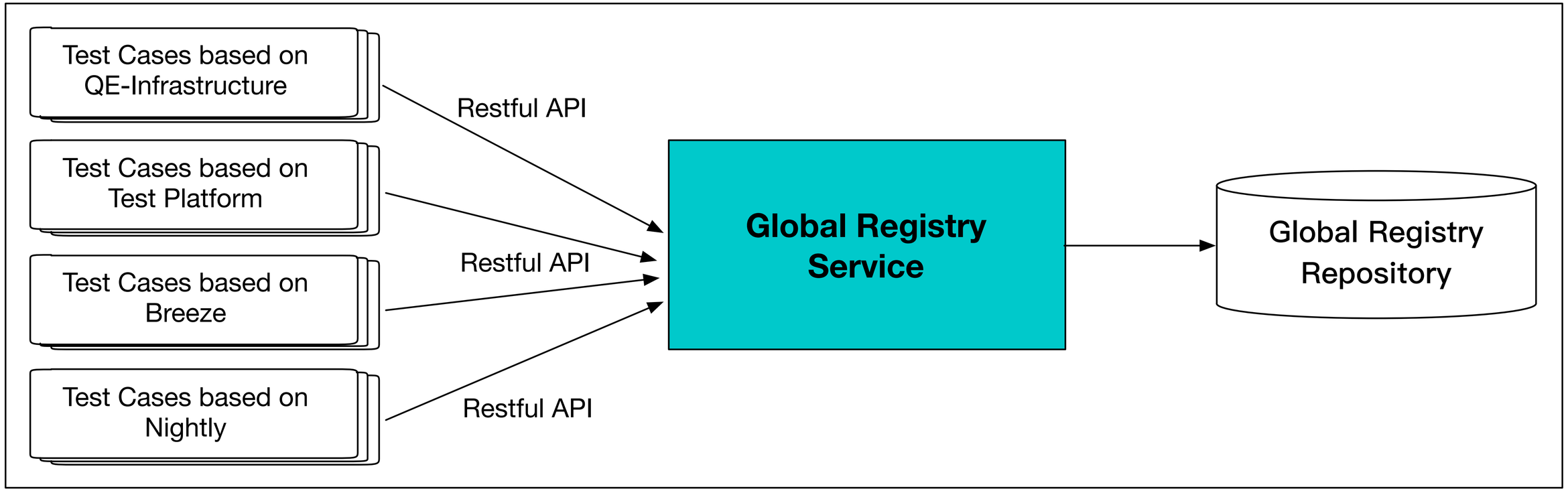
Global Registry Repository

```
shstoreId=1
defaultLocale=en-US
defaultCurrency=USD
supportedLocales=en-US,es-MX
supportedCurrencies=USD
defaultWebTLD=com
```

After

```
public static String getCurrencyCode() {
  return GlobalRegistry.byCountry(GlobalEnvironment.getCountry()).getDefaultCurrency();
}
```

Global Registry Service



My column in Software Testing

My WeChat