ThoughtWorks® IT大咖说 知识分享平台

契约测试

微服务独立交付的金钥匙

ThoughtWorks® | IT大咖说 知识分享平台

微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，每个服务运行在其独立的进程中，服务间采用轻量级的通信机制互相沟通（通常是基于HTTP协议的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被独立的部署到生产环境、类生产环境等。

http://martinfowler.com/articles/microservices.html

微服务架构是一种架构模式，它提倡将单一应用程序**划分成一组小的服务**，每个服务运行在其**独立的进程**中，服务间采用**轻量级的通信机制**互相沟通（通常是基于HTTP协议的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被**独立的部署**到生产环境、类生产环境等。

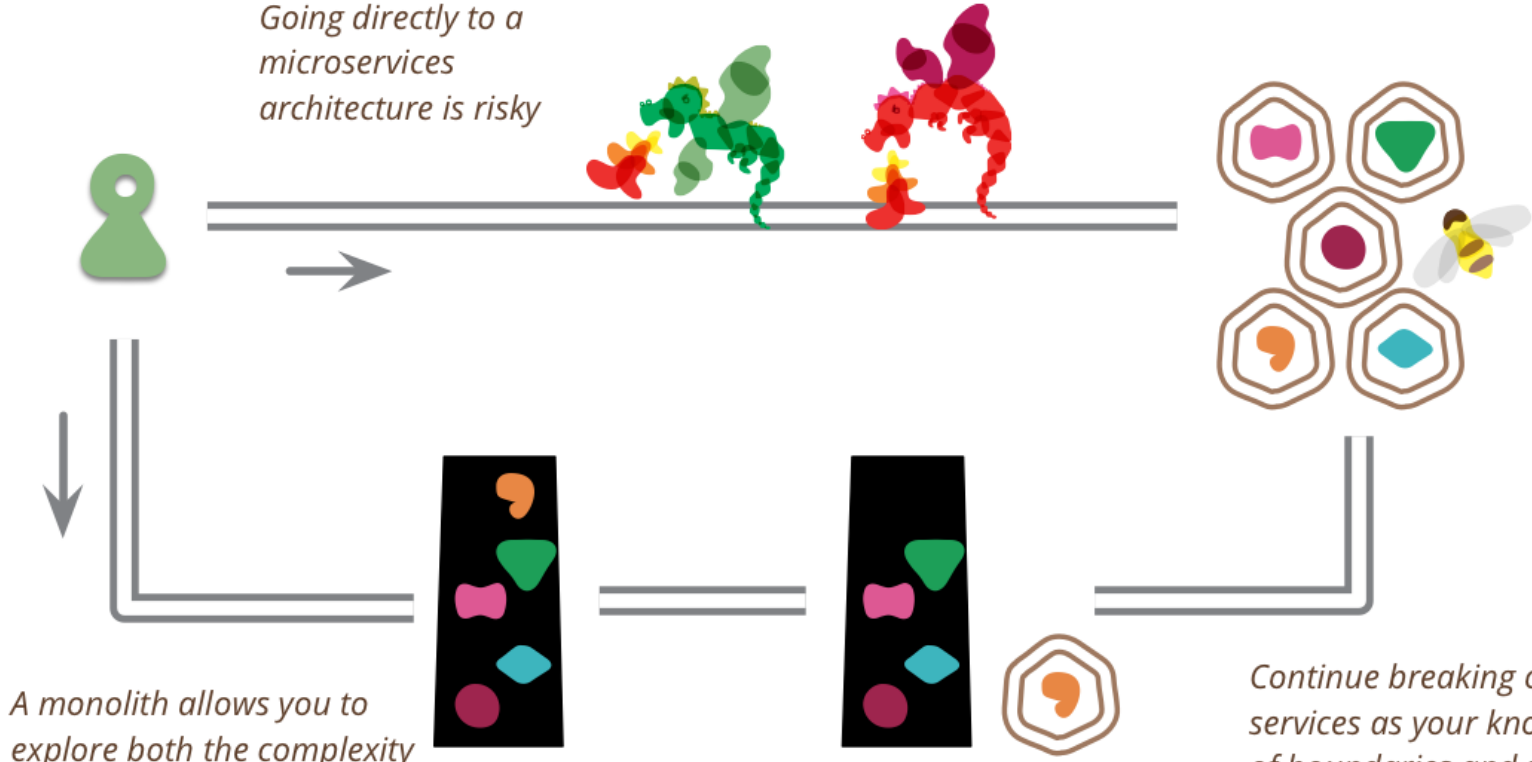
http://martinfowler.com/articles/microservices.html

ThoughtWorks® | IT大咖说 知识分享平台

微服务架构是一种架构模式，它提倡将单一应用程序划分成一组小的服务，每个服务运行在其独立的进程中，服务间采用轻量级的通信机制互相沟通（通常是基于HTTP协议的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被**独立的部署**到生产环境、类生产环境等。

http://martinfowler.com/articles/microservices.html

- 组件化
- 弹性架构
- 去中心化

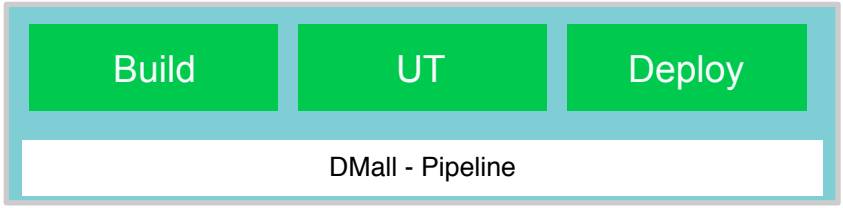

Going directly to a microservices architecture is risky
A monolith allows you to explore both the complexity of a system and its component boundaries
As complexity rises start breaking out some microservices
Continue breaking out services as your knowledge of boundaries and service management increases

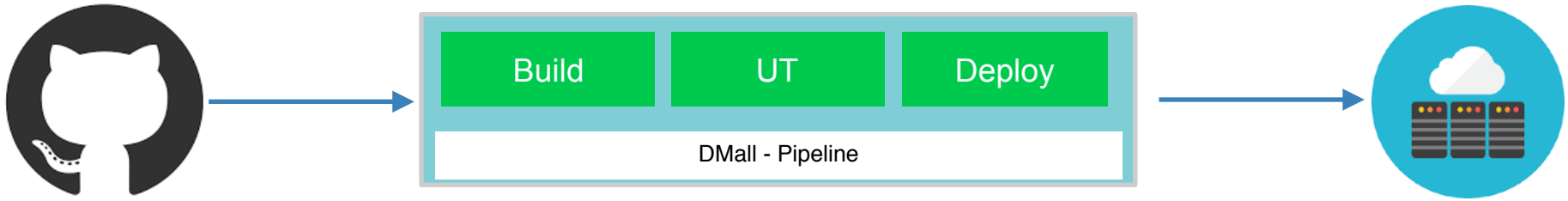
ThoughtWorks® | IT大咖说 知识分享平台

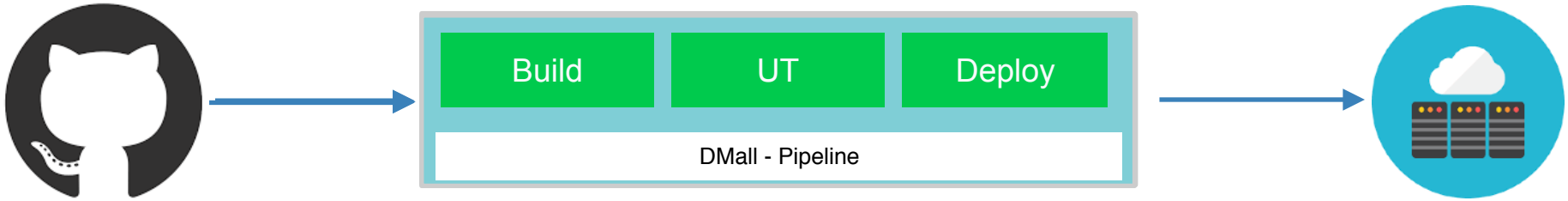
怎么说呢？

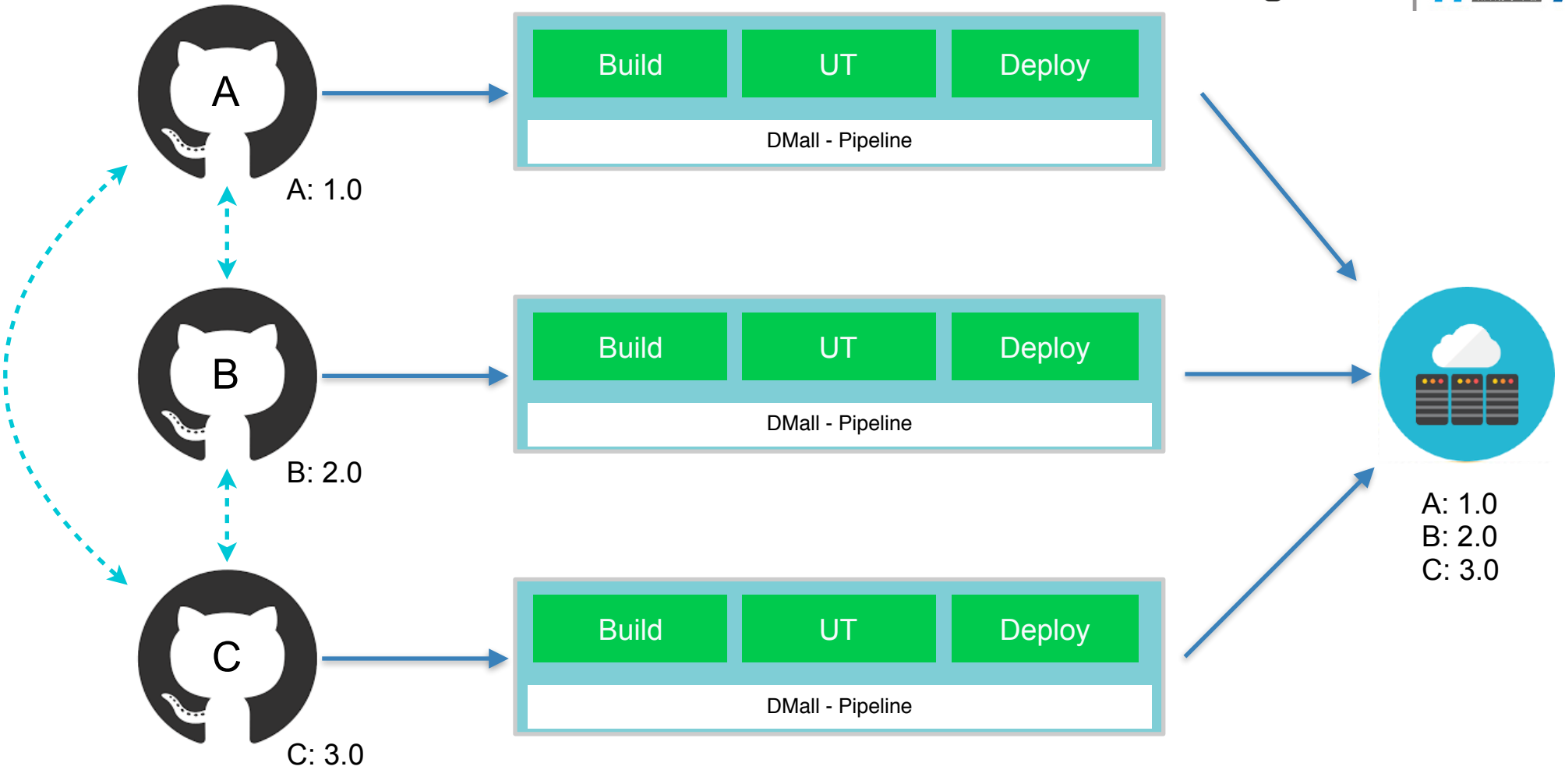
这样，我给你举个栗子吧！

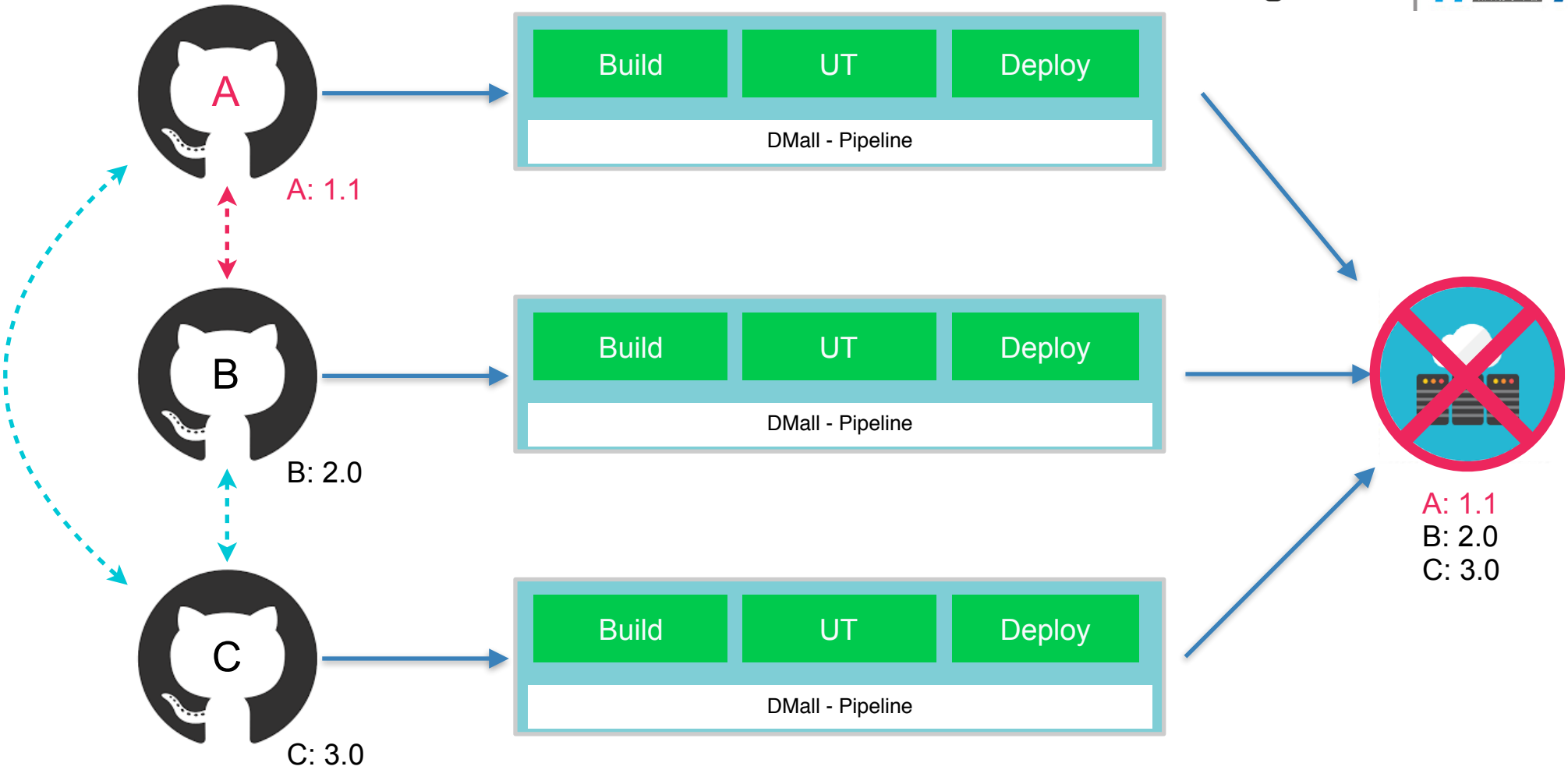

ONLINE
SHOPPING

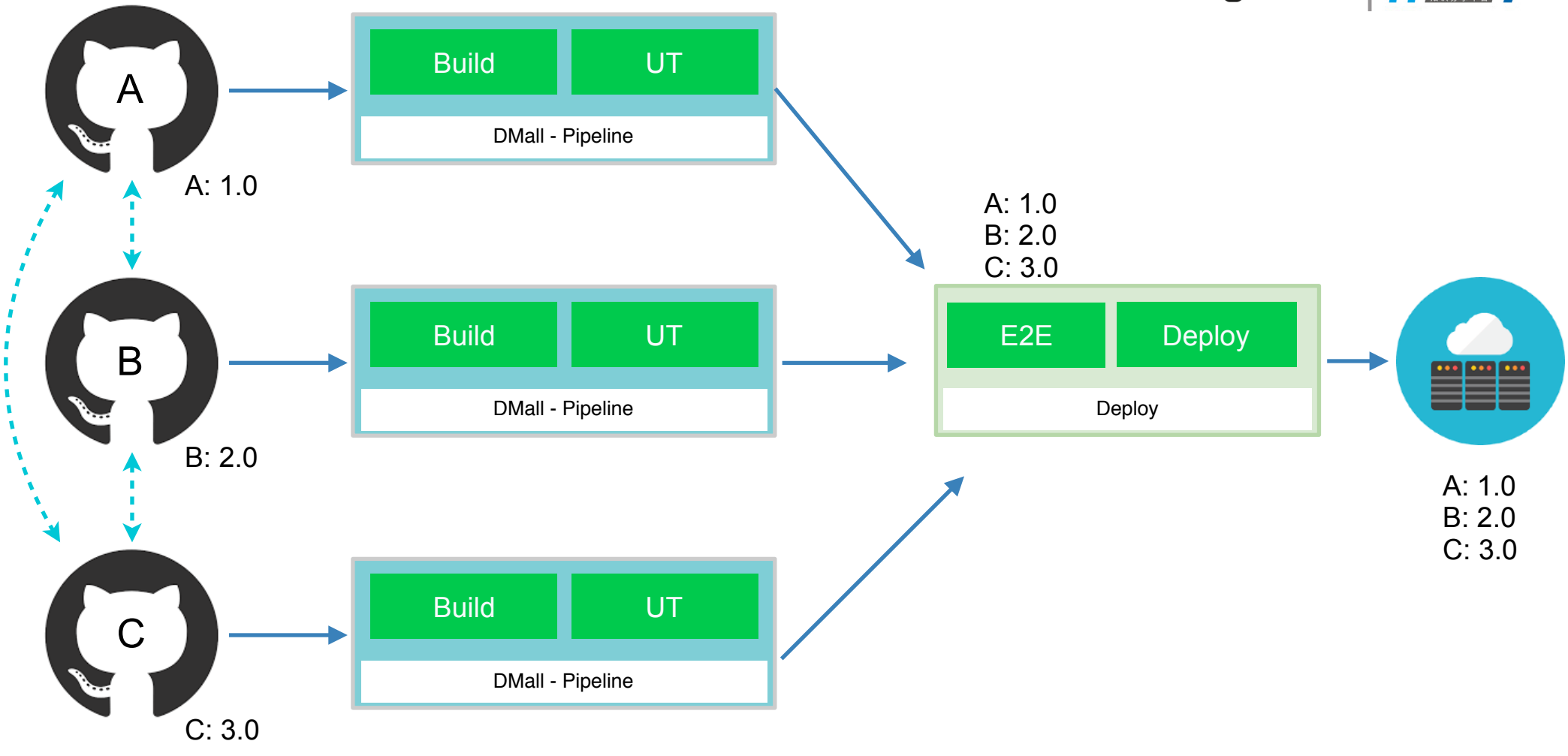

Build
UT
Deploy
DMall - Pipeline

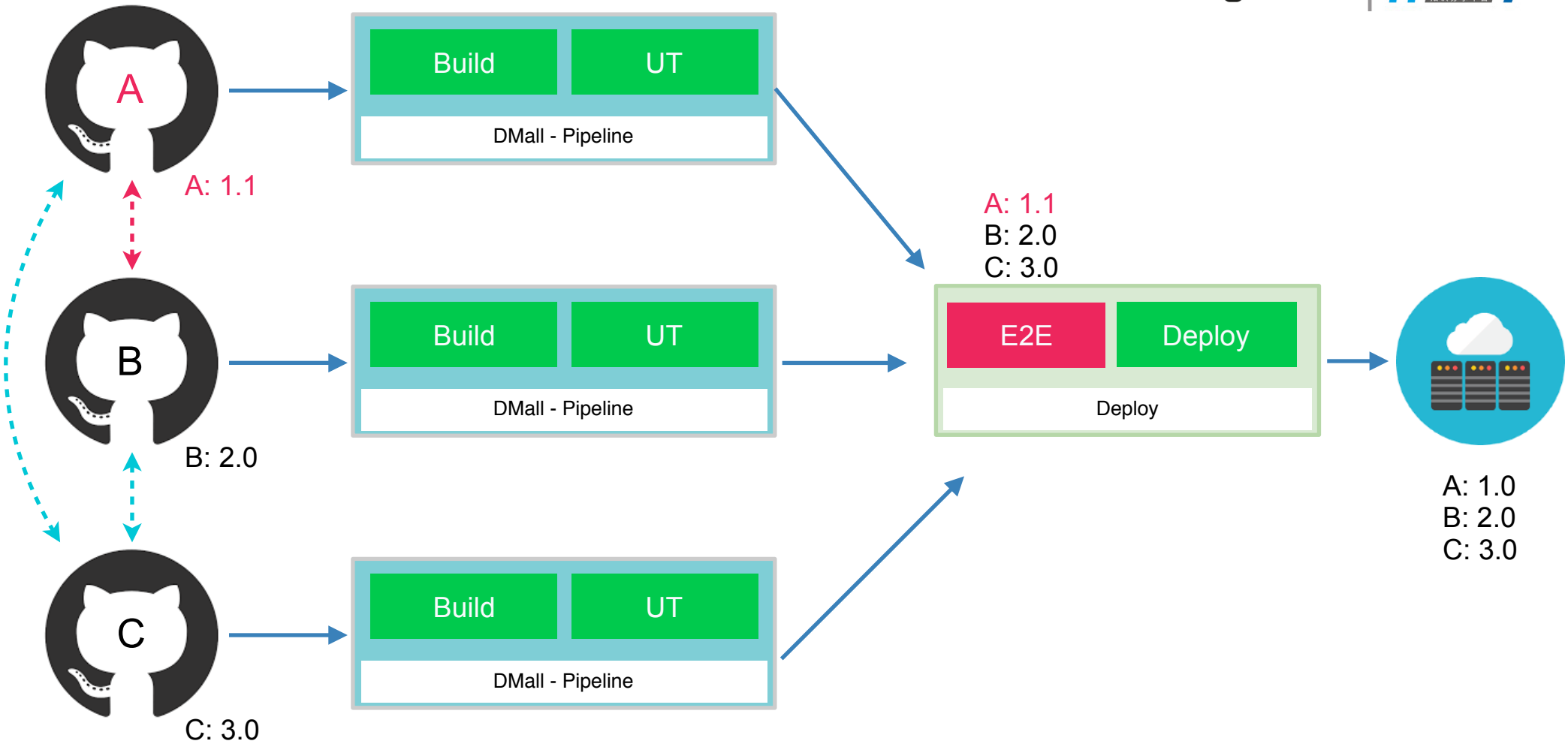

Build
UT
Deploy
DMall - Pipeline

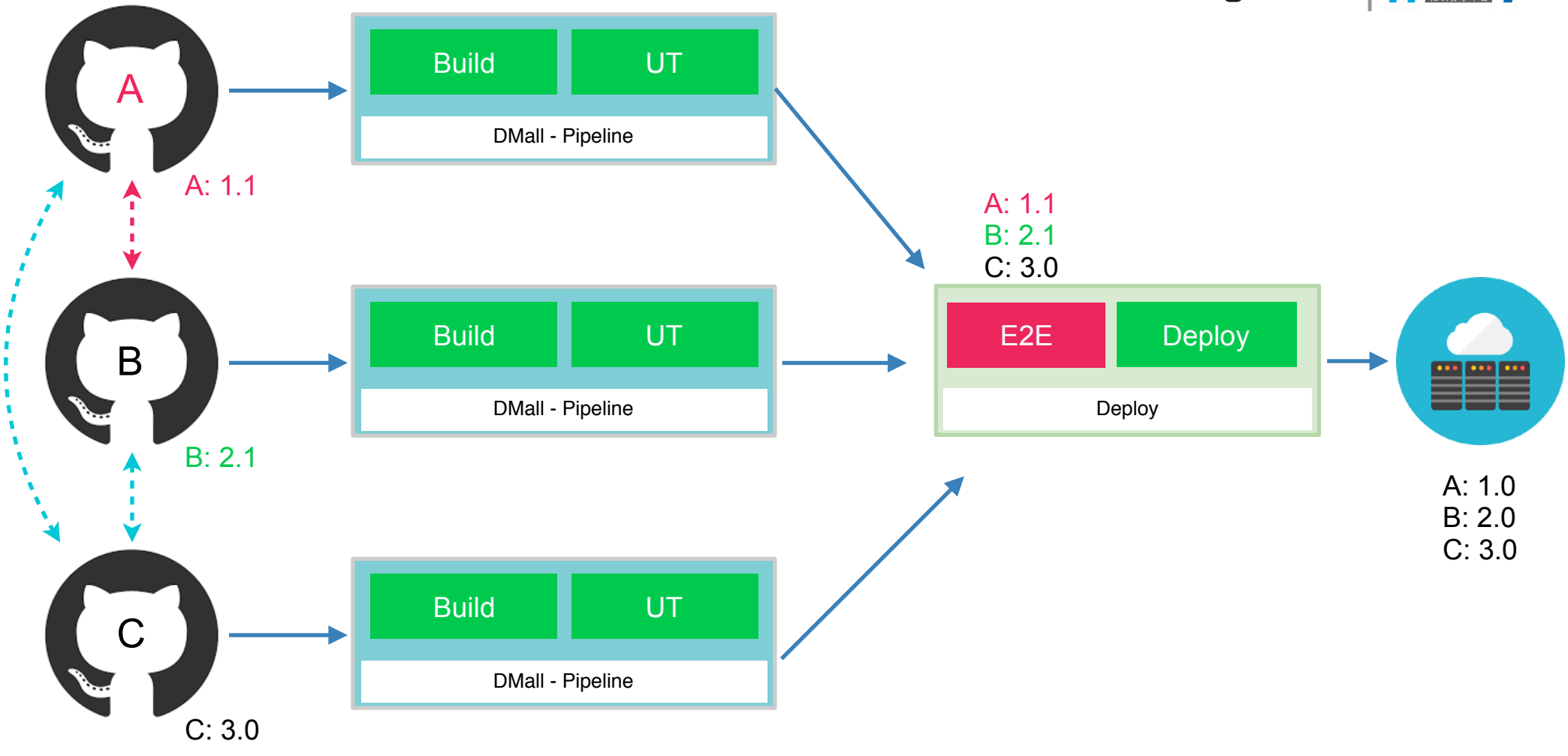

Build
UT
Deploy
DMall - Pipeline

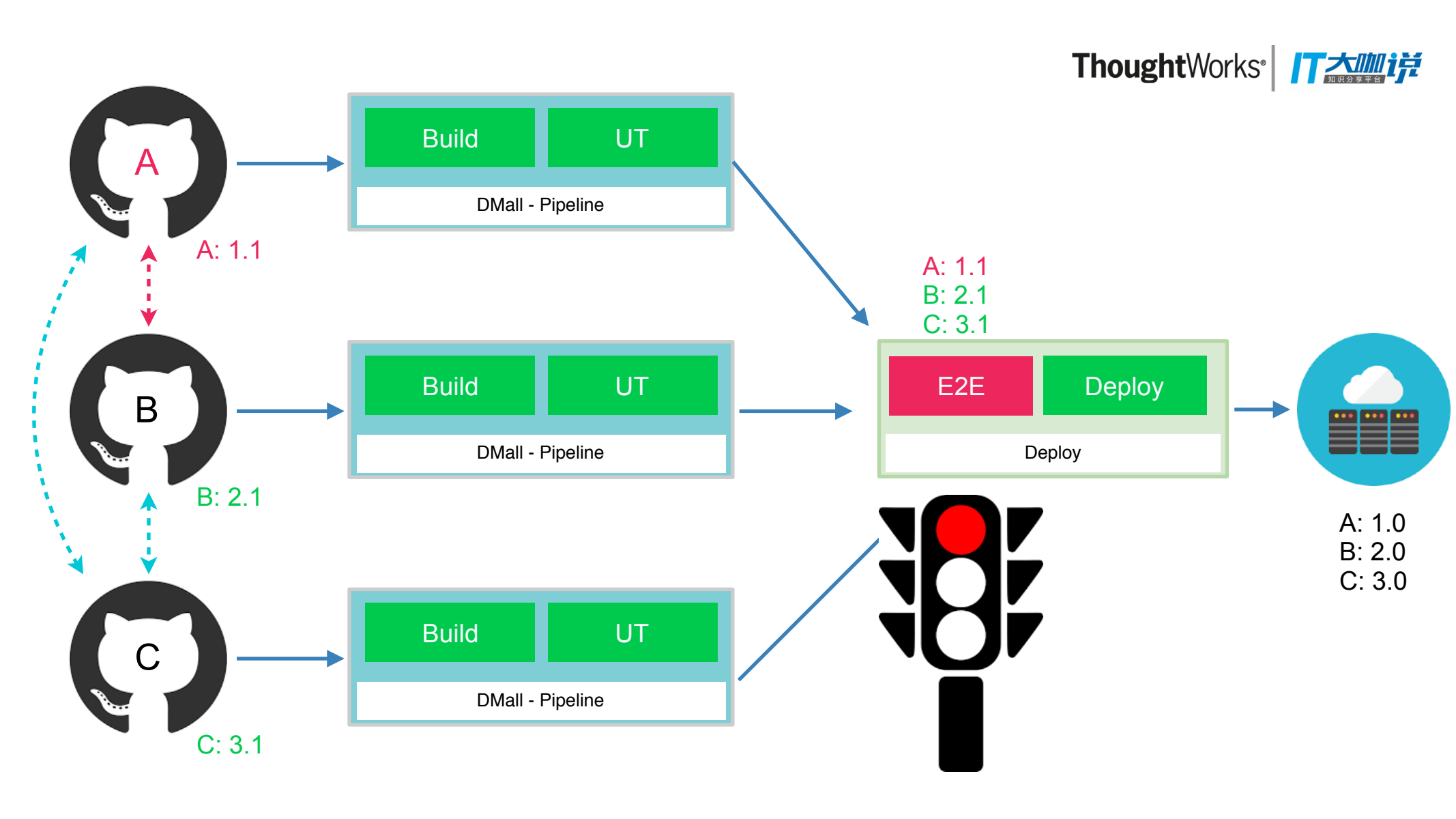

A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
Deploy
DMall - Pipeline
Build
UT
Deploy
DMall - Pipeline
Build
UT
Deploy
DMall - Pipeline
A: 1.0
B: 2.0
C: 3.0

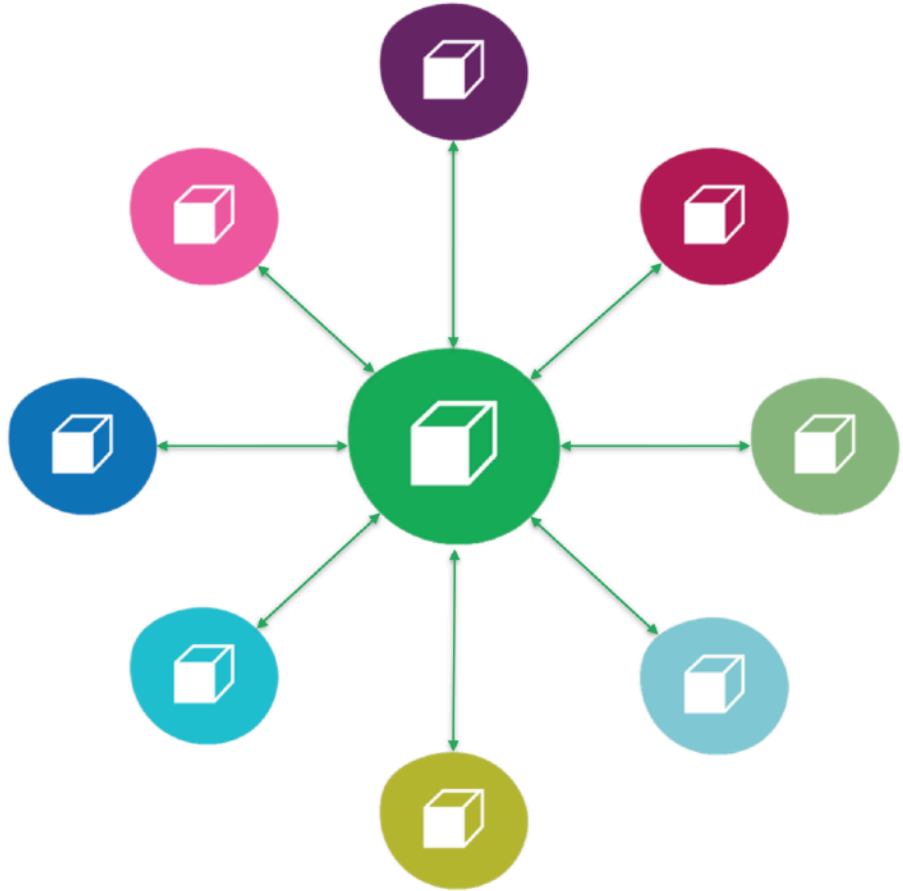

A
A: 1.1
B
B: 2.0
C
C: 3.0
Build
UT
Deploy
DMall - Pipeline
Build
UT
Deploy
DMall - Pipeline
Build
UT
Deploy
DMall - Pipeline
A: 1.1
B: 2.0
C: 3.0

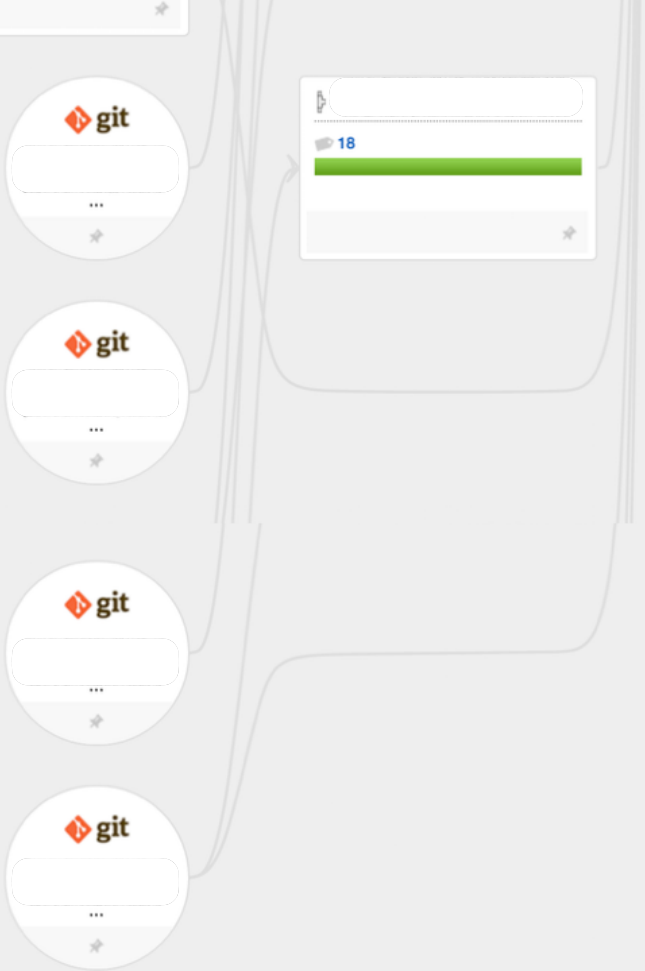

A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
A: 1.0
B: 2.0
C: 3.0
E2E
Deploy
Deploy
A: 1.0
B: 2.0
C: 3.0

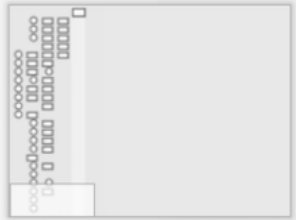

A
A: 1.1
B
B: 2.0
C
C: 3.0
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
A: 1.1
B: 2.0
C: 3.0
E2E
Deploy
Deploy
A: 1.0
B: 2.0
C: 3.0

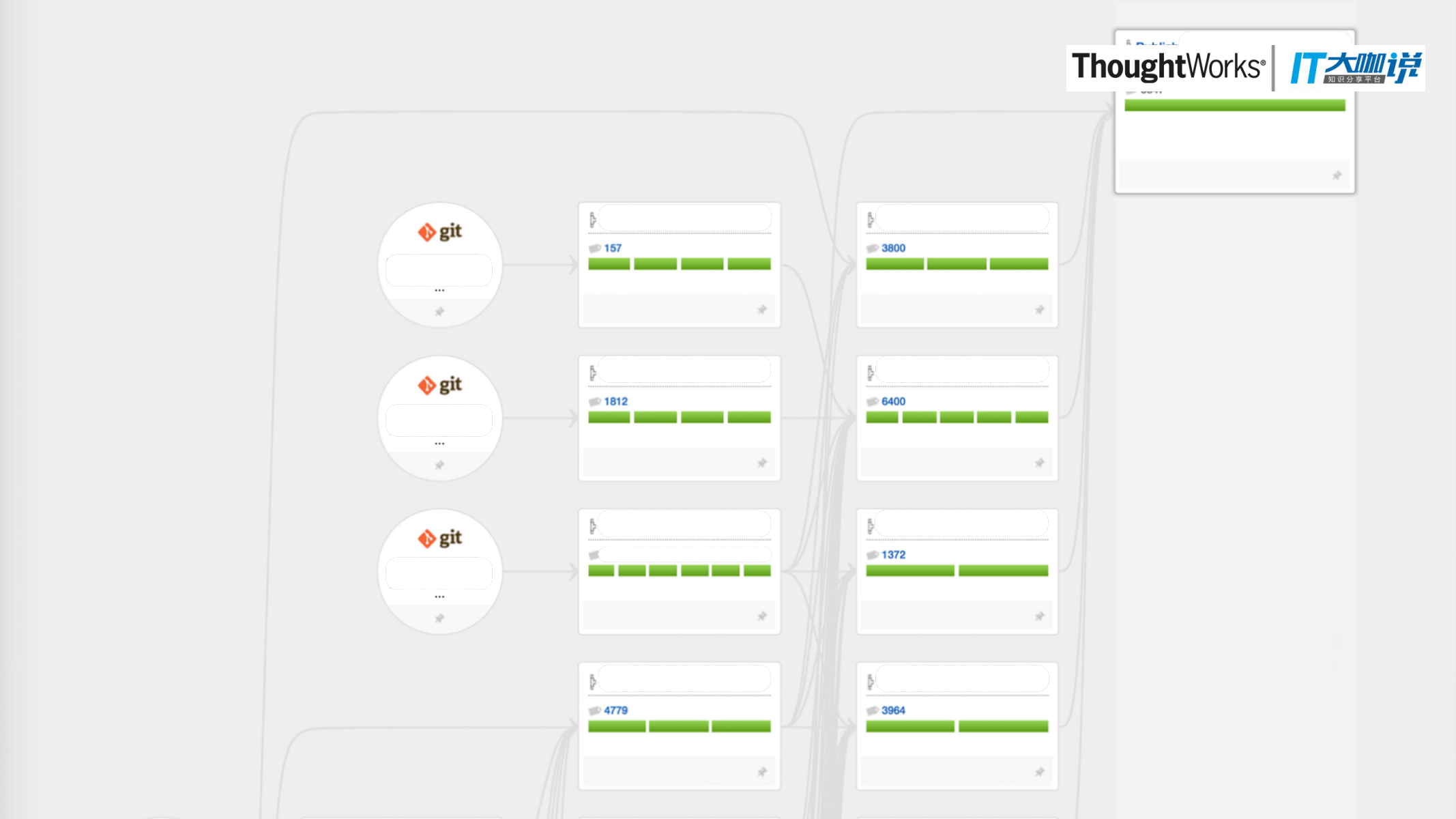

A
A: 1.1
B
B: 2.1
C
C: 3.0
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
A: 1.1
B: 2.1
C: 3.0
E2E
Deploy
Deploy
A: 1.0
B: 2.0
C: 3.0


A
A: 1.1
B
B: 2.1
C
C: 3.1
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
A: 1.1
B: 2.1
C: 3.1
E2E
Deploy
Deploy
A: 1.0
B: 2.0
C: 3.0



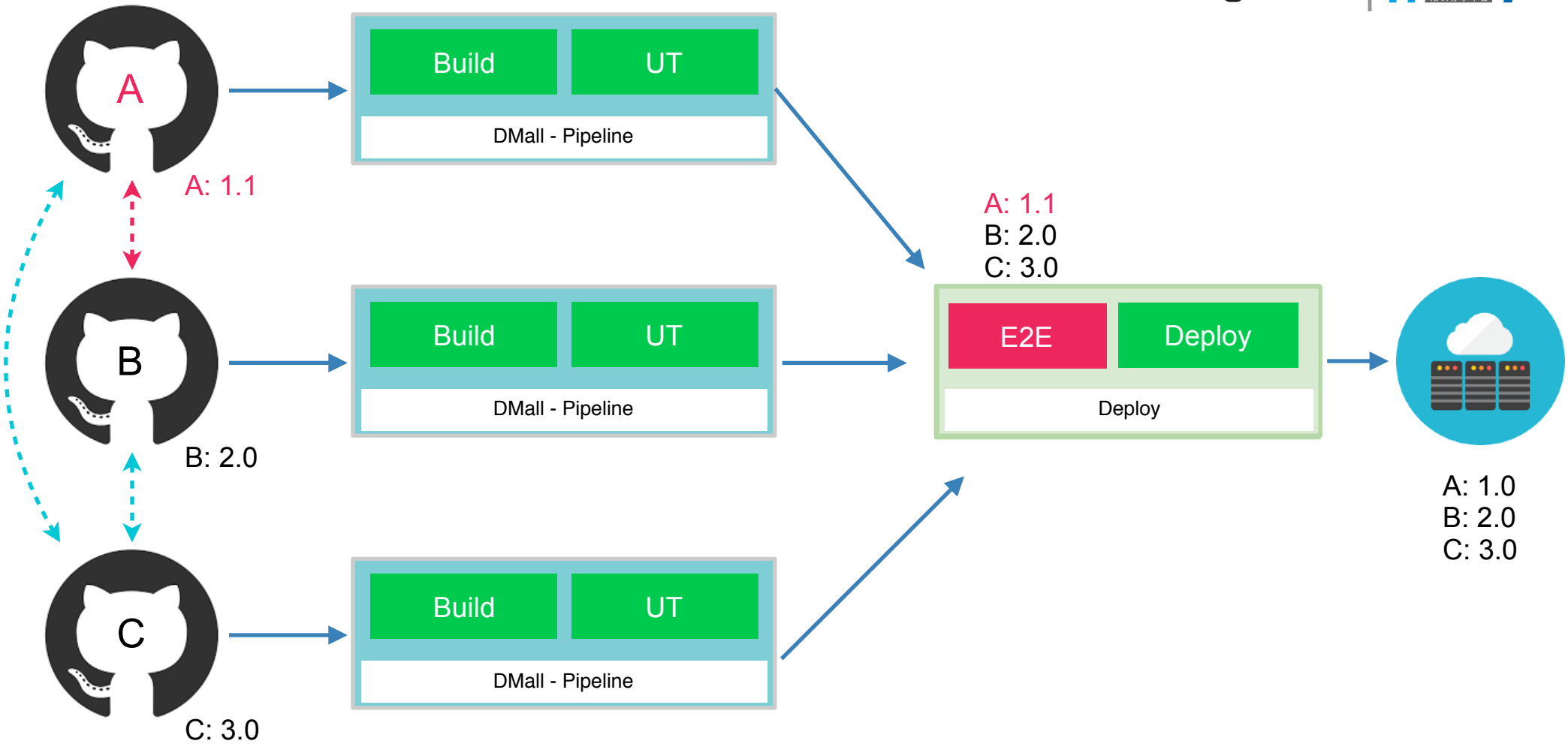


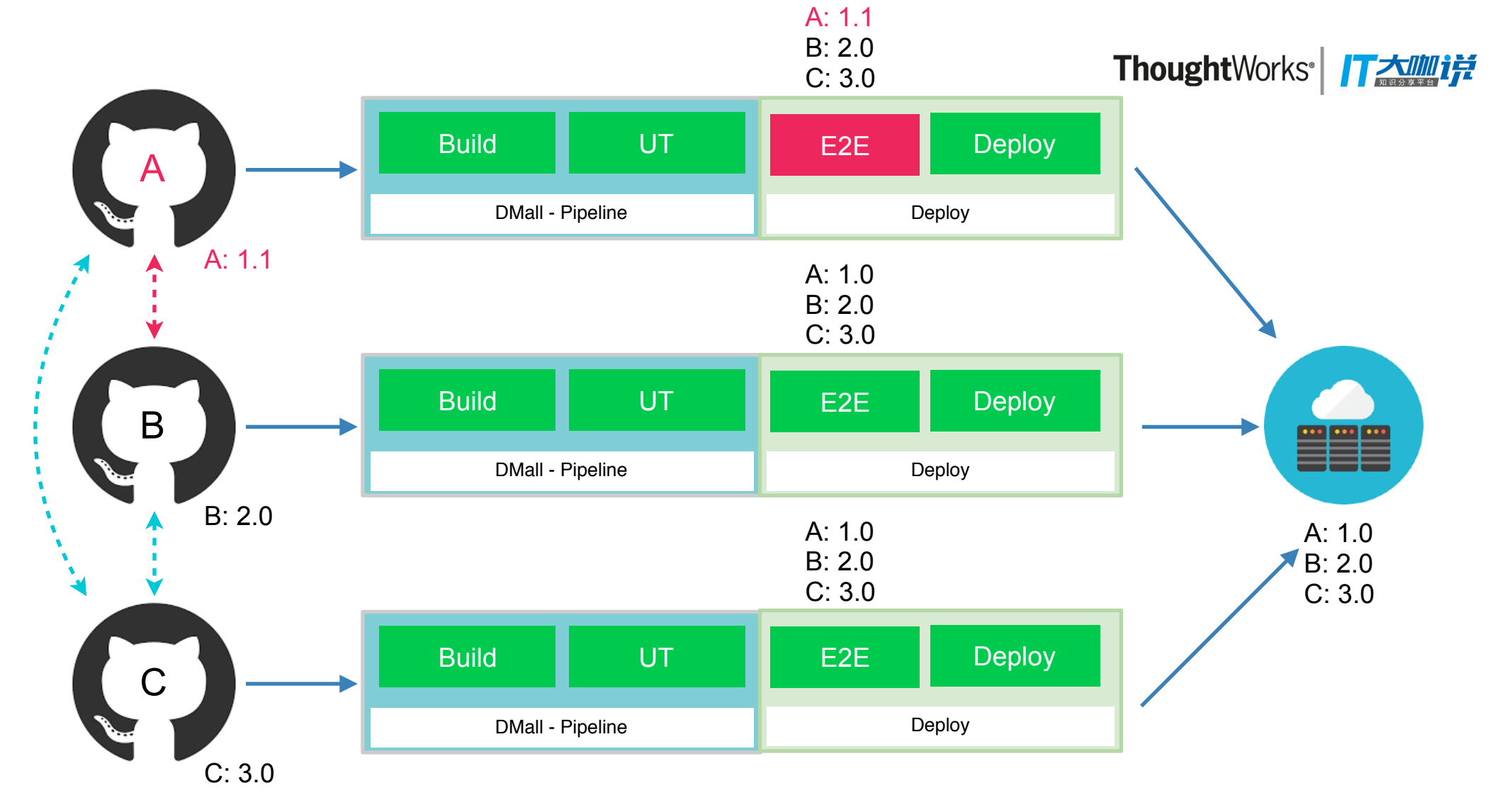


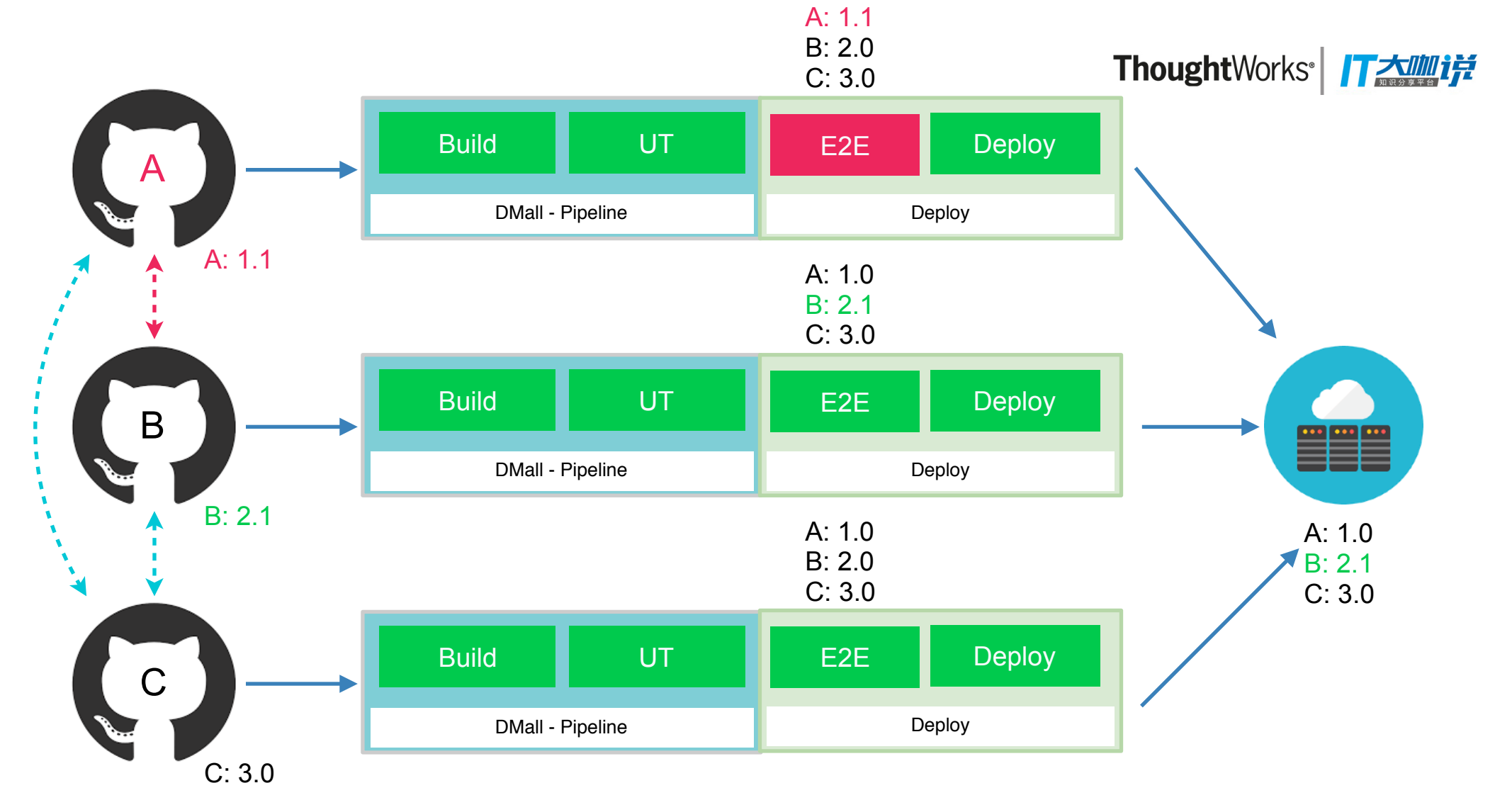

ThoughtWorks
IT大咖说
知识分享平台
git
...
157
3800
git
...
1812
6400
git
...
1372
4779
3964

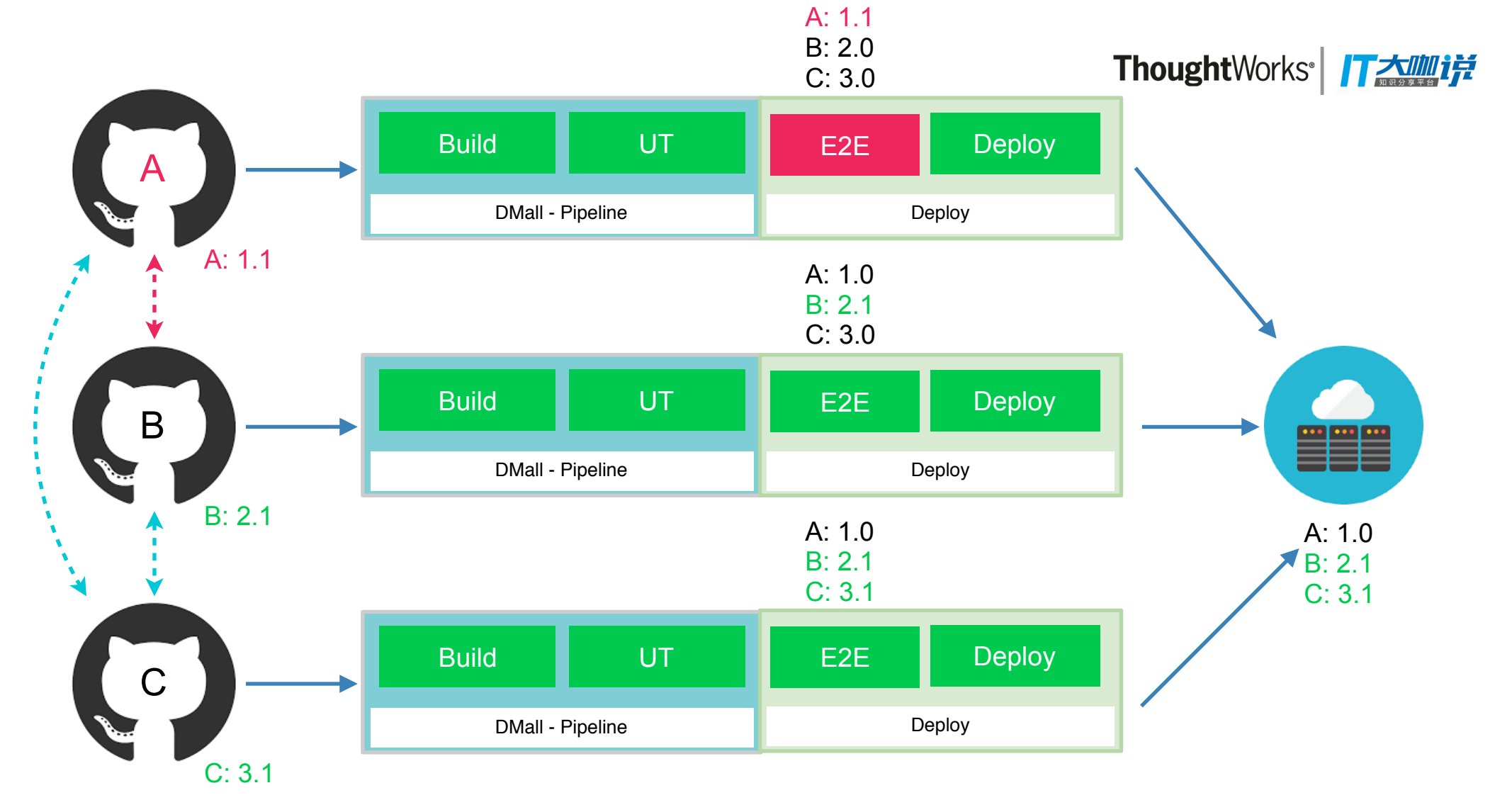
ThoughtWorks® | IT大咖说 知识分享平台

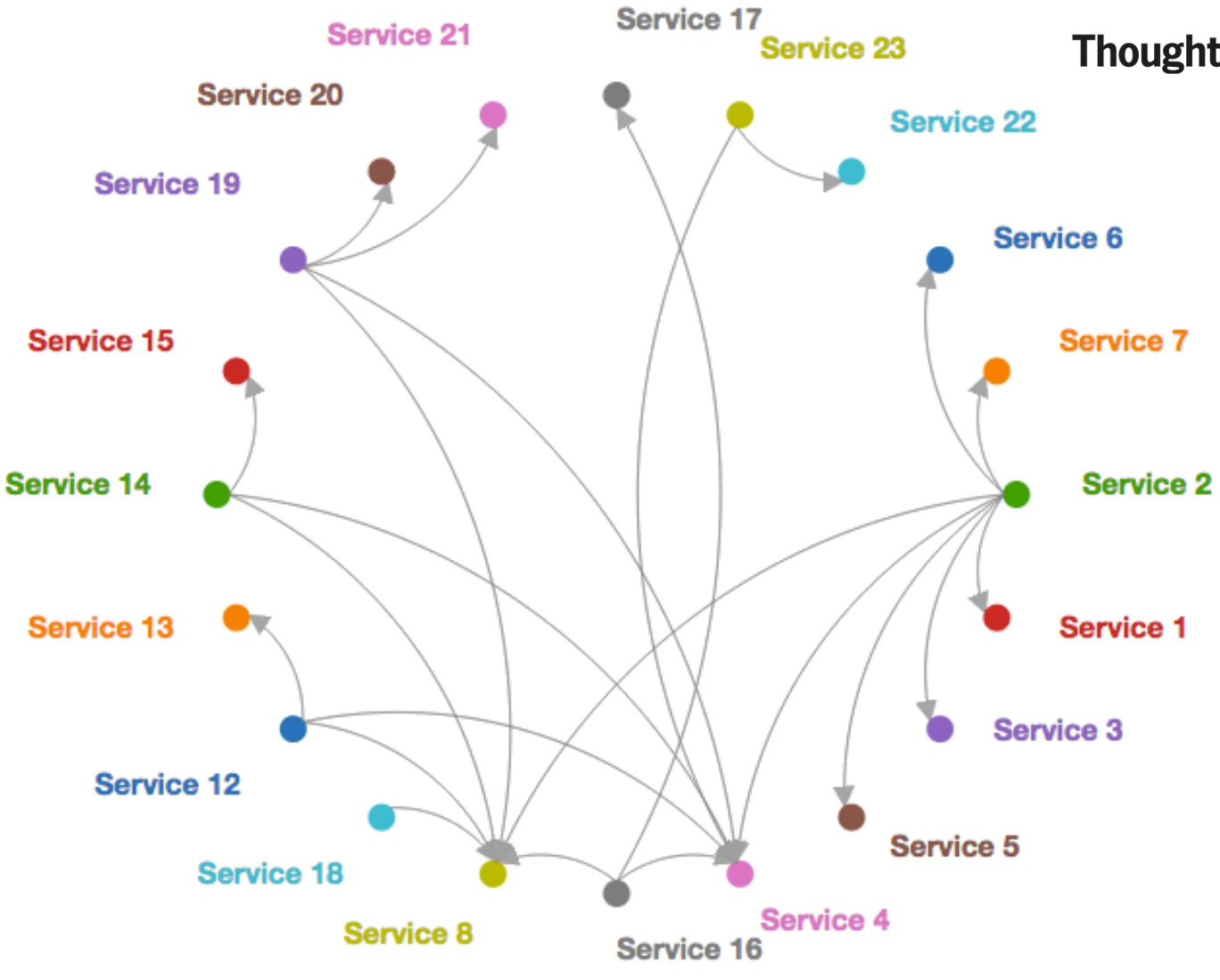

ThoughtWorks
IT大咖说
知识分享平台

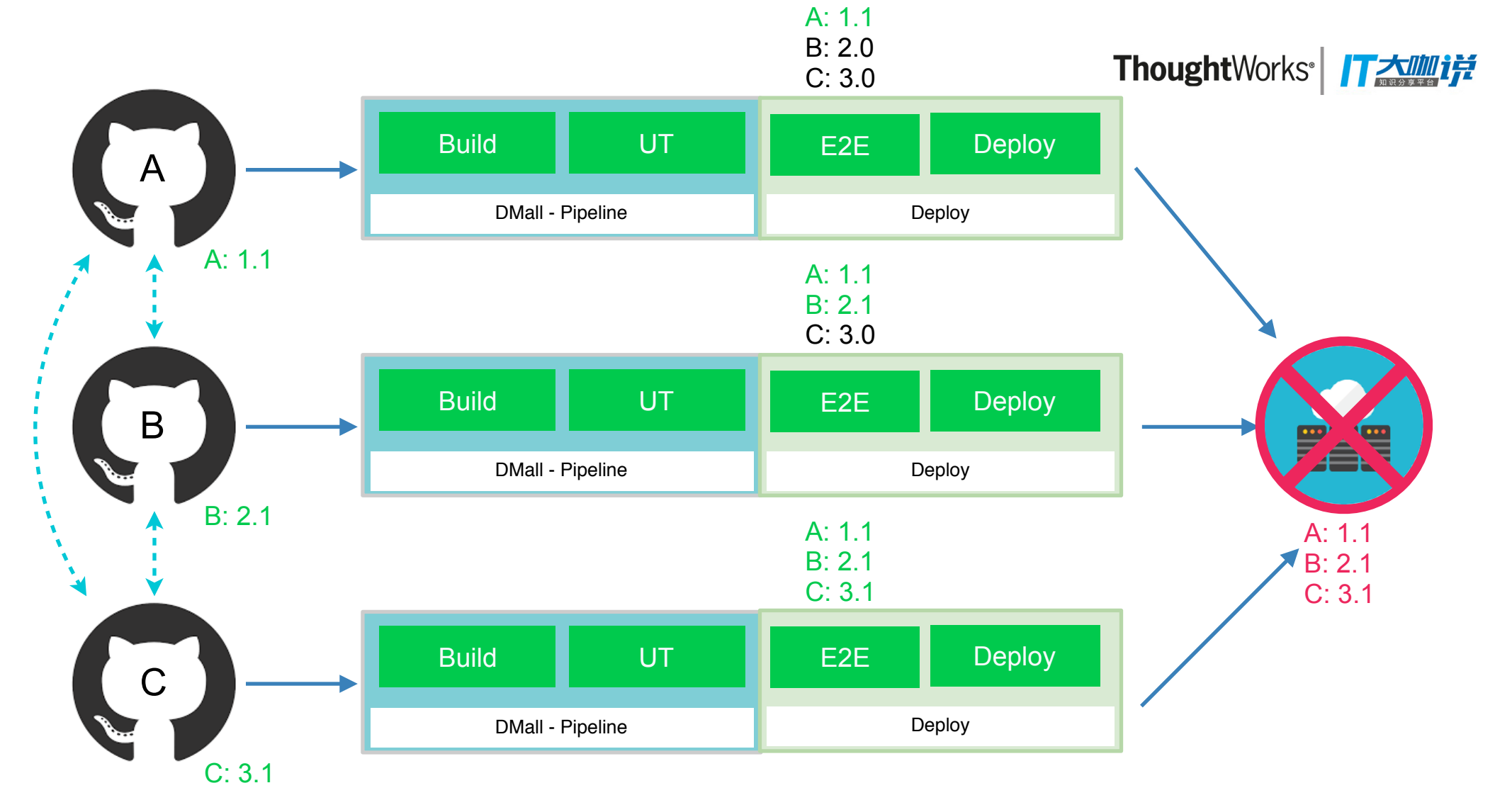

A
A: 1.1
B
B: 2.0
C
C: 3.0
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
Build
UT
DMall - Pipeline
A: 1.1
B: 2.0
C: 3.0
E2E
Deploy
Deploy
A: 1.0
B: 2.0
C: 3.0


A: 1.1
B: 2.0
C: 3.0
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0
B
B: 2.0
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0
C
C: 3.0
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0


A: 1.1
B: 2.0
C: 3.0
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.1
C: 3.0
B
B: 2.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0
C
C: 3.0
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.1
C: 3.0

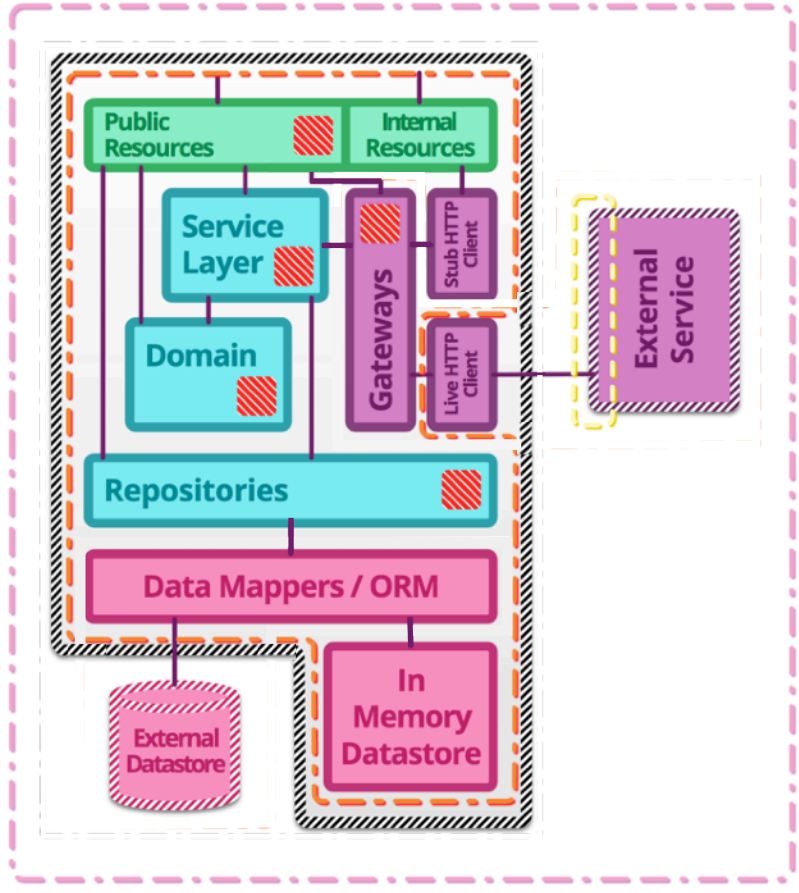

A: 1.1
B: 2.0
C: 3.0
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.1
C: 3.0
B
B: 2.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.1
C: 3.1
A: 1.0
B: 2.1
C: 3.1
C
C: 3.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy




A: 1.1
B: 2.0
C: 3.0
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.1
B: 2.1
C: 3.0
B
B: 2.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.1
B: 2.1
C: 3.1
C
C: 3.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.1
B: 2.1
C: 3.1


ThoughtWorks
IT大咖说
知识分享平台

TESTING STRATEGY

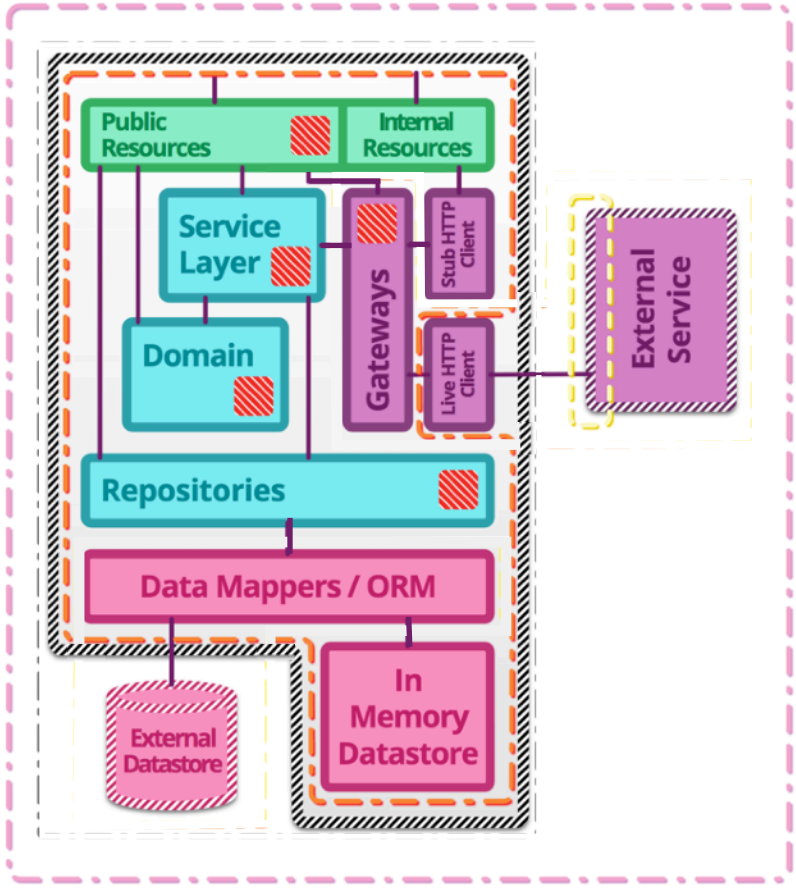


End-to-end tests : verify that a system meets external requirements and achieves its goals, testing the entire system, from end to end.

Contract tests : verify interactions at the boundary of an external service asserting that it meets the contract expected by a consuming service.

Component tests : limit the scope of the exercised software to a portion of the system under test, manipulating the system through internal code interfaces and using test doubles to isolate the code under test from other components.

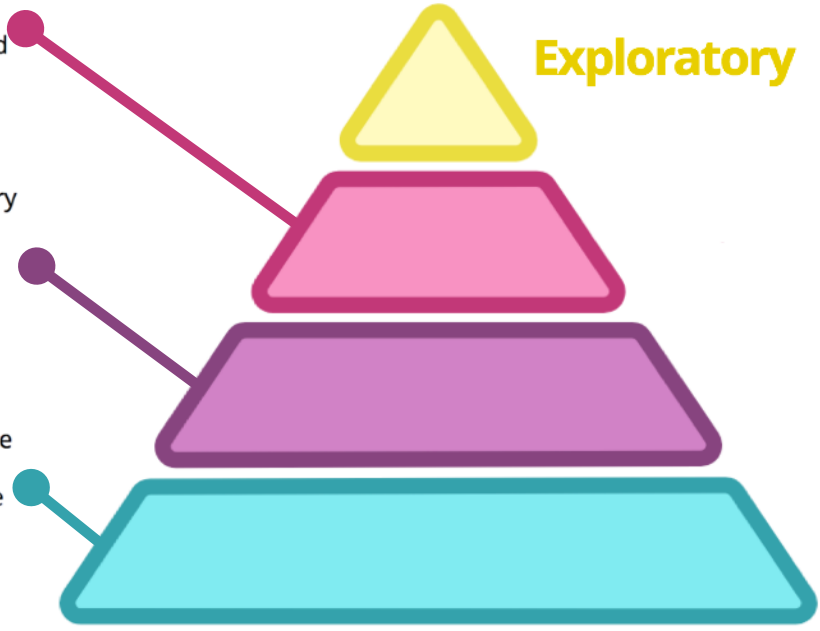
TESTING STRATEGY



End-to-end tests : verify that a system meets external requirements and achieves its goals, testing the entire system, from end to end.

Contract tests : verify interactions at the boundary of an external service asserting that it meets the contract expected by a consuming service.

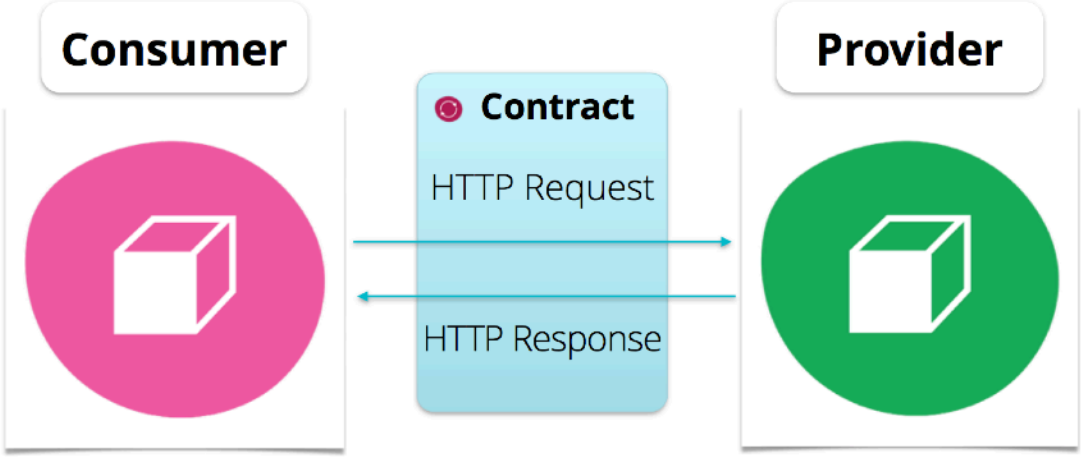
Component tests : limit the scope of the exercised software to a portion of the system under test, manipulating the system through internal code interfaces and using test doubles to isolate the code under test from other components.




ThoughtWorks®
IT大咖说
知识分享平台

Contract Testing

什么是Contract Testing



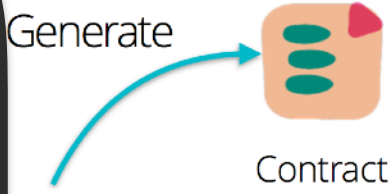
Consumer：Service的使用者，向provider发起HTTP请求来获取数据；

Provider：Service的提供者，接收consumer的HTTP请求并返回数据；

Contract：契约，一种定义在consumer与provider之间的交互方式；

Contract Testing 验证Provider是否按照期望的方式与Consumer进行交互。

```
PactNet.Mocks.MockHttpService.Models;
espace myHR.Tests.Pacts

public class myCoreUserContactsPacts : UserApiPactsBase<ContactsFixture>
{
    [Fact]
    public async Task should_get_user_contacts()
    {
       mockmyCoreApi
         .Given("user has a contact")
         .UponReceiving("a GET request for user contacts")
         .With(
                new ProviderServiceRequest
                {
                    Method = HttpVerb.Get,
                    Path = $"/users/{id}/contacts",
                })
          .WillRespondWith(
                new ProviderServiceResponse
                {
                    Status = 200,
                    Body = new[]
                    {
                        new
                        {
                            name = "Zhang BA",
                        }
                    }
                });
        await myCoreApiClient.GetContacts(id);
        mockmyCoretApi.VerifyInteractions();
    }
}
public class ContactsFixture : ServiceFixture
{
    public ContactsFixture()
    {
        mockService("myHR", "myCoreApi.Contacts", 10853);
    }
}
```

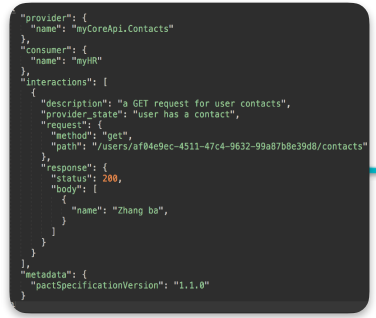


Pact

Provider

```
PactNet.Mocks.MockHttpService.Models;
espace myHR.Tests.Pacts

public class myCoreUserContactsPacts : UserApiPactsBase<ContactsFixture>
{
    [Fact]
    public async Task should_get_user_contacts()
    {
       mockmyCoreApi
         .Given("user has a contact")
         .UponReceiving("a GET request for user contacts")
         .With(
                new ProviderServiceRequest
                {
                    Method = HttpVerb.Get,
                    Path = $"/users/{id}/contacts",
                })
          .WillRespondWith(
                new ProviderServiceResponse
                {
                    Status = 200,
                    Body = new[]
                    {
                        new
                        {
                            name = "Zhang BA",
                        }
                    }
                });
        await myCoreApiClient.GetContacts(id);
        mockmyCoretApi.VerifyInteractions();
    }
}
public class ContactsFixture : ServiceFixture
{
    public ContactsFixture()
    {
        mockService("myHR", "myCoreApi.Contacts", 10853);
    }
}
```

```
{
  "provider": {
    "name": "myCoreApi.Contacts"
  },
  "consumer": {
    "name": "myHR"
  },
  "interactions": [
    {
      "description": "a GET request for user contacts",
      "provider_state": "user has a contact",
      "request": {
        "method": "get",
        "path": "/users/af04e9ec-4511-47c4-9632-99a87b8e39d8/contacts"
      },
      "response": {
        "status": 200,
        "body": [
          {
            "name": "Zhang ba",
          }
        ]
      }
    }
  ],
  "metadata": {
    "pactSpecificationVersion": "1.1.0"
  }
}
```



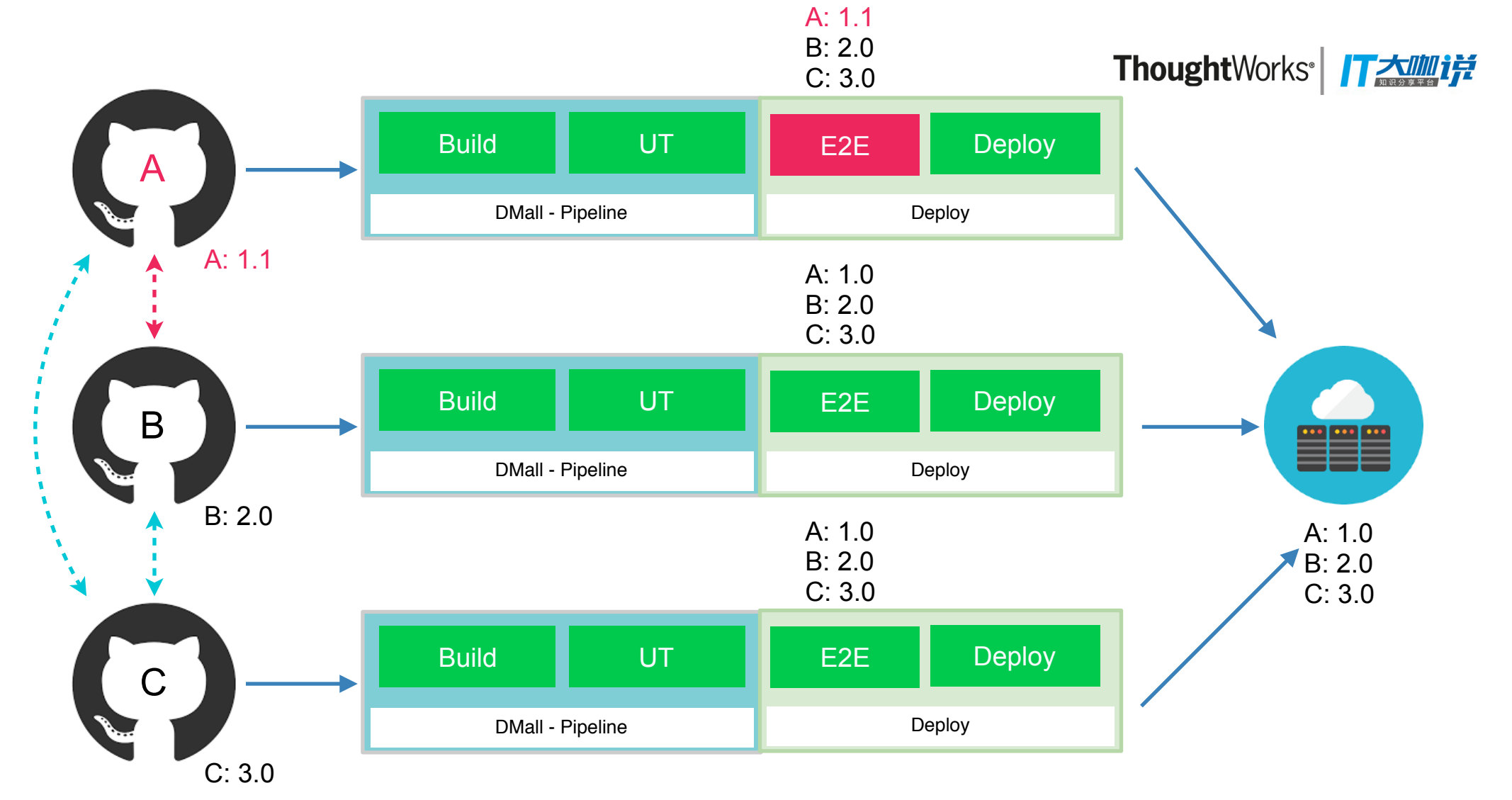
Retrieve

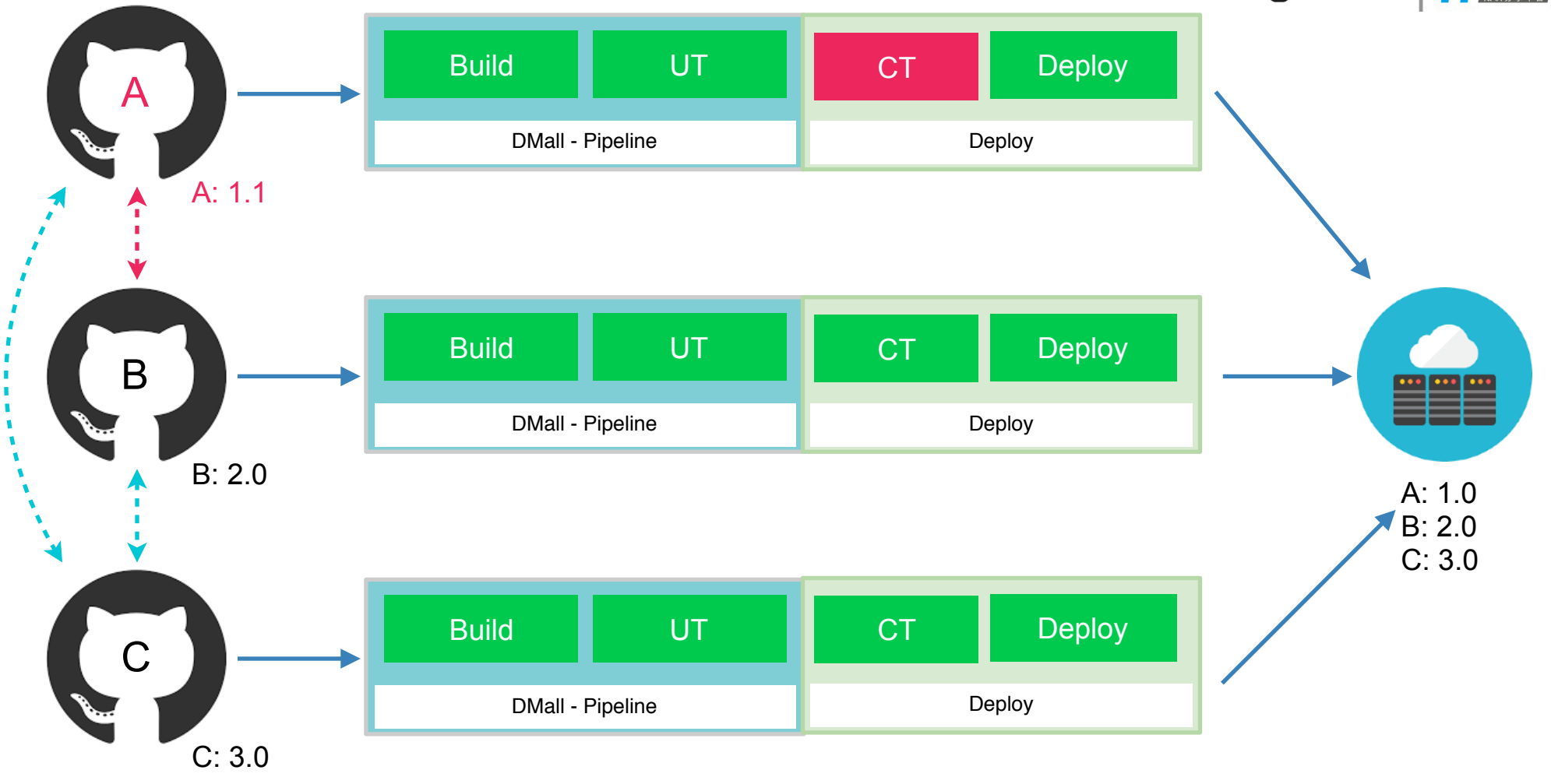
Pact

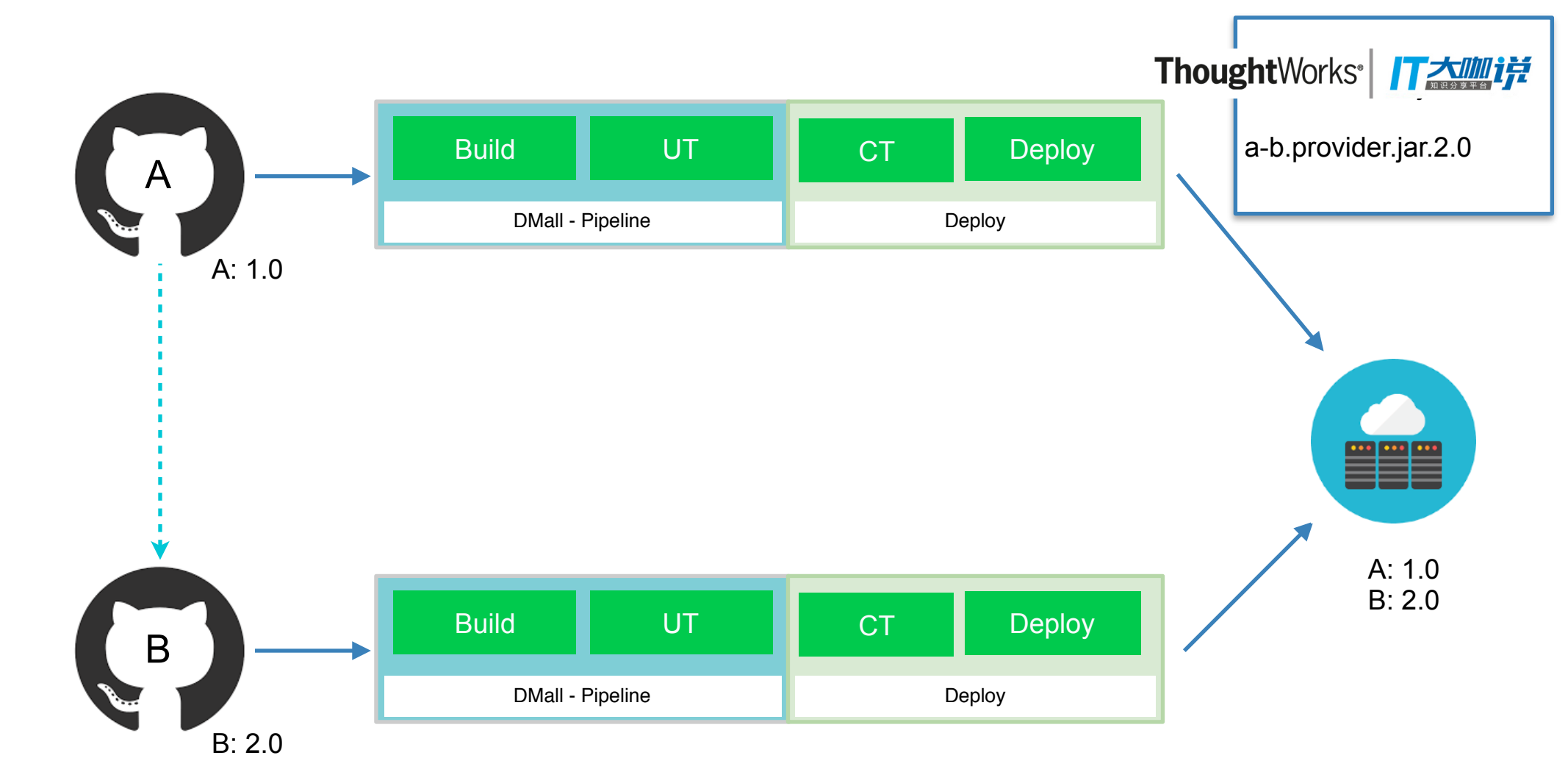
Verify

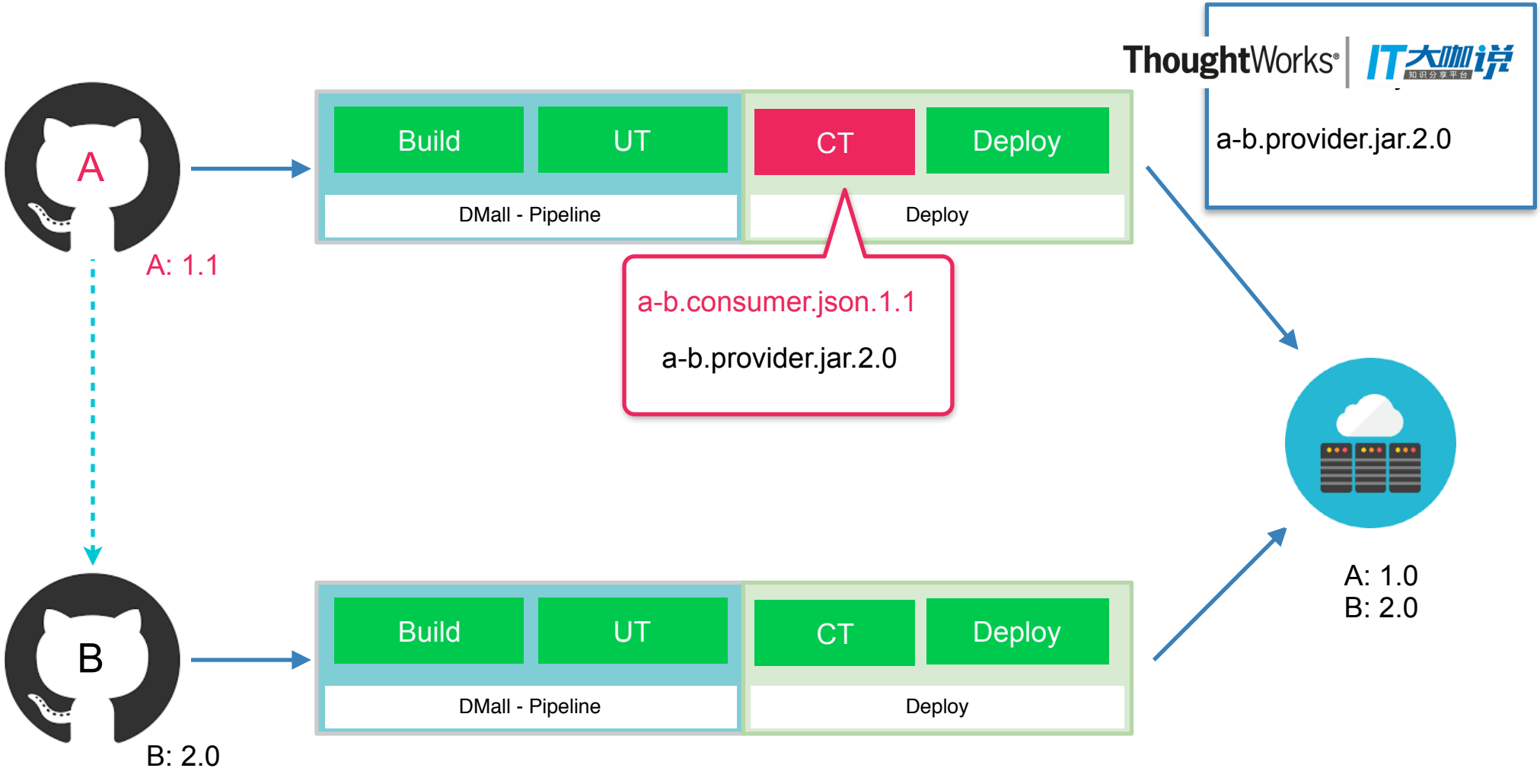
Consumer

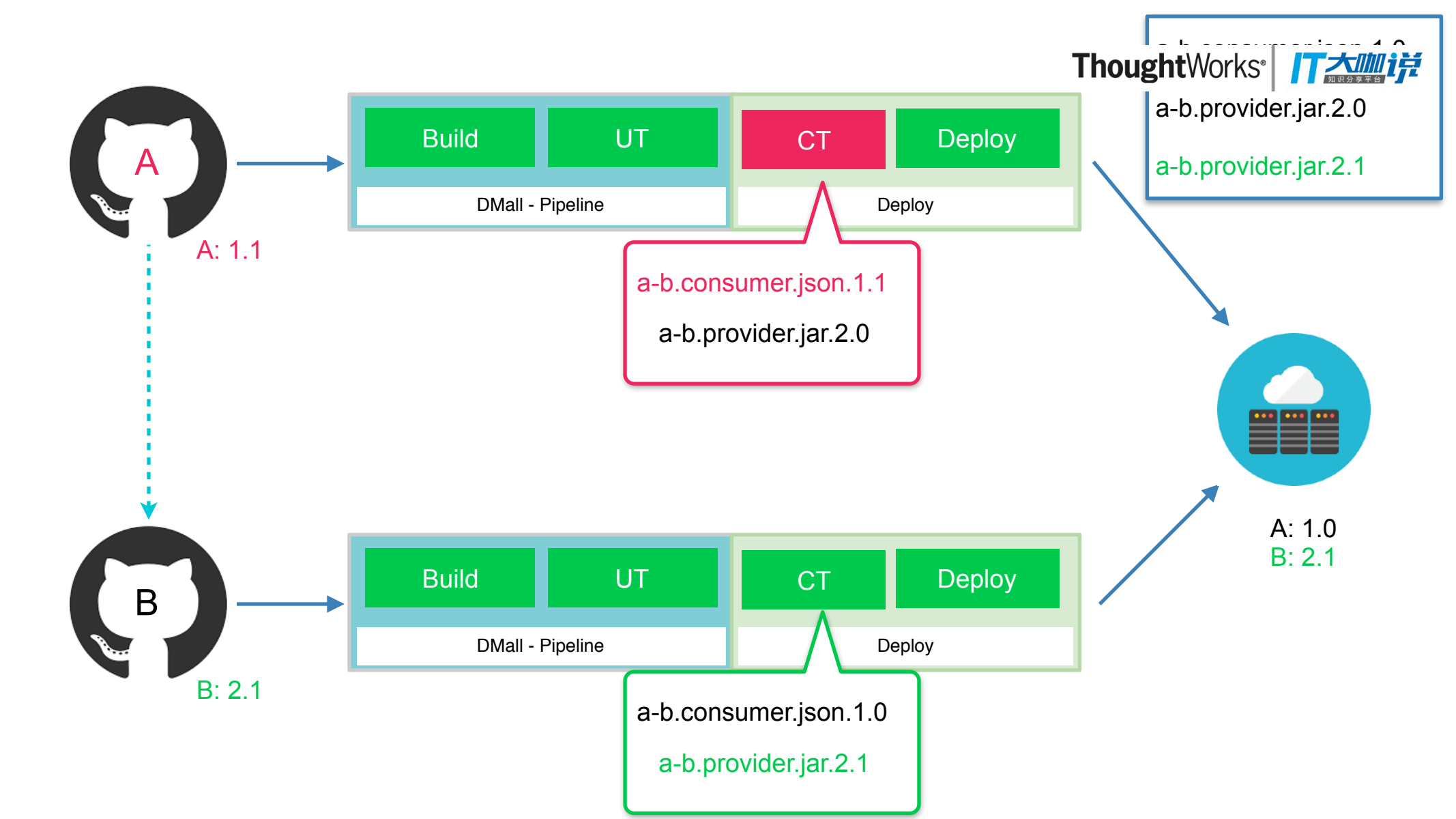
```
ng PactNet;
amespace myCore.Contract.Test.By.Consumer.myHR
{
    public class ContactsFacts : ContractFactsBase
    {
        [Fact]
        public void should_verify_contract_with_myHR()
        {
            var pactVerifier = new PactVerifier();

            pactVerifier.ProviderStatesFor("myHR")
                .ProviderState("user has a contact", Setup);

            pactVerifier.ServiceProvider("myCoreApi.Contacts")
                .HonoursPactWith("myHR")
                .PactUri("Pacts/myHR/myHR-myCore-api.contacts.json")
                .Verify();
        }

        void Setup()
        {
            var user = new User(1, "Zhang BA");
            Scope.Resolve<IRepository<User>>().Update(user);
            UseStubAccountService(true);
        }
    }
}
```

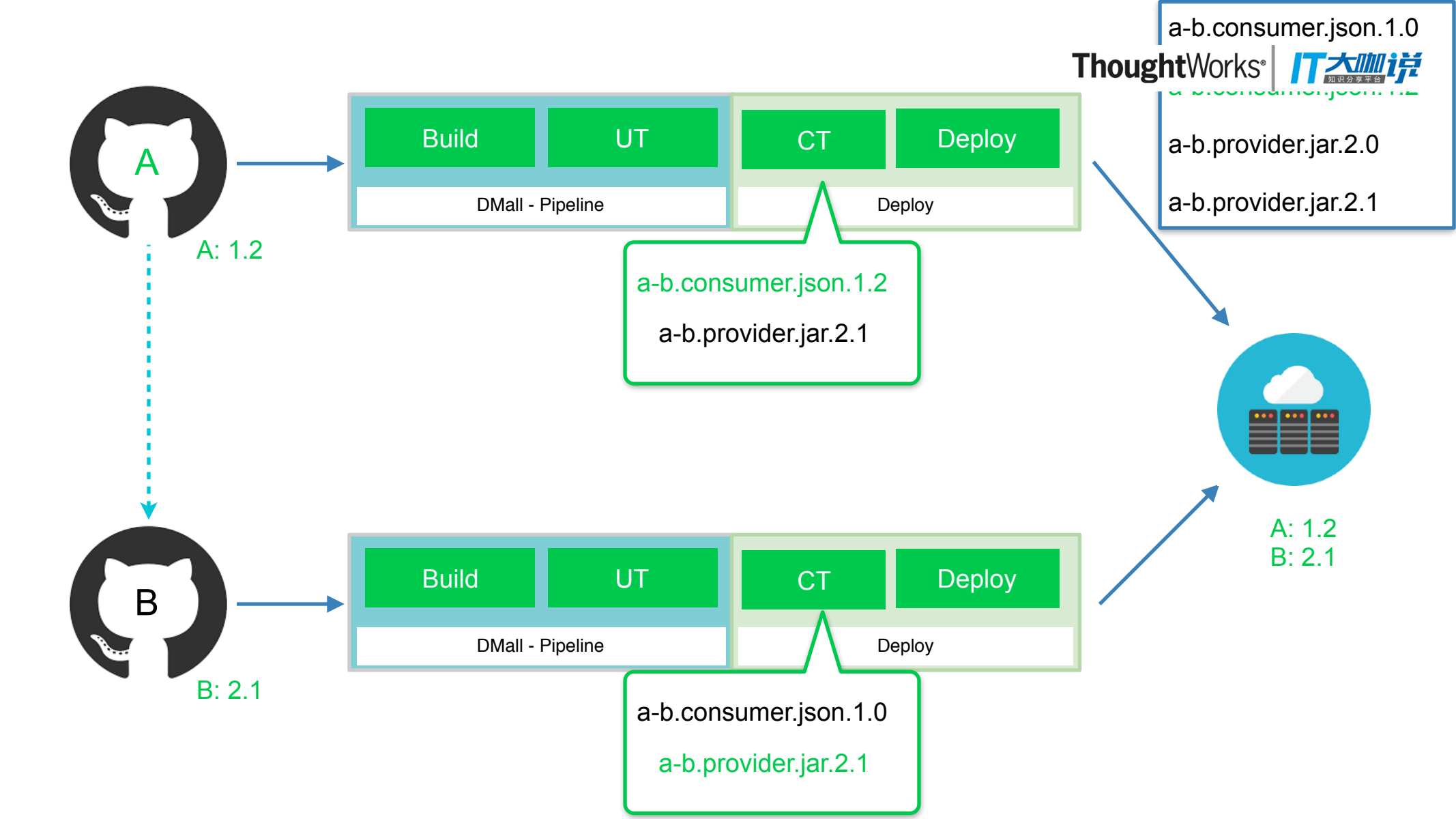

A: 1.1
B: 2.0
C: 3.0
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0
B
B: 2.0
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0
C
C: 3.0
Build
UT
E2E
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0

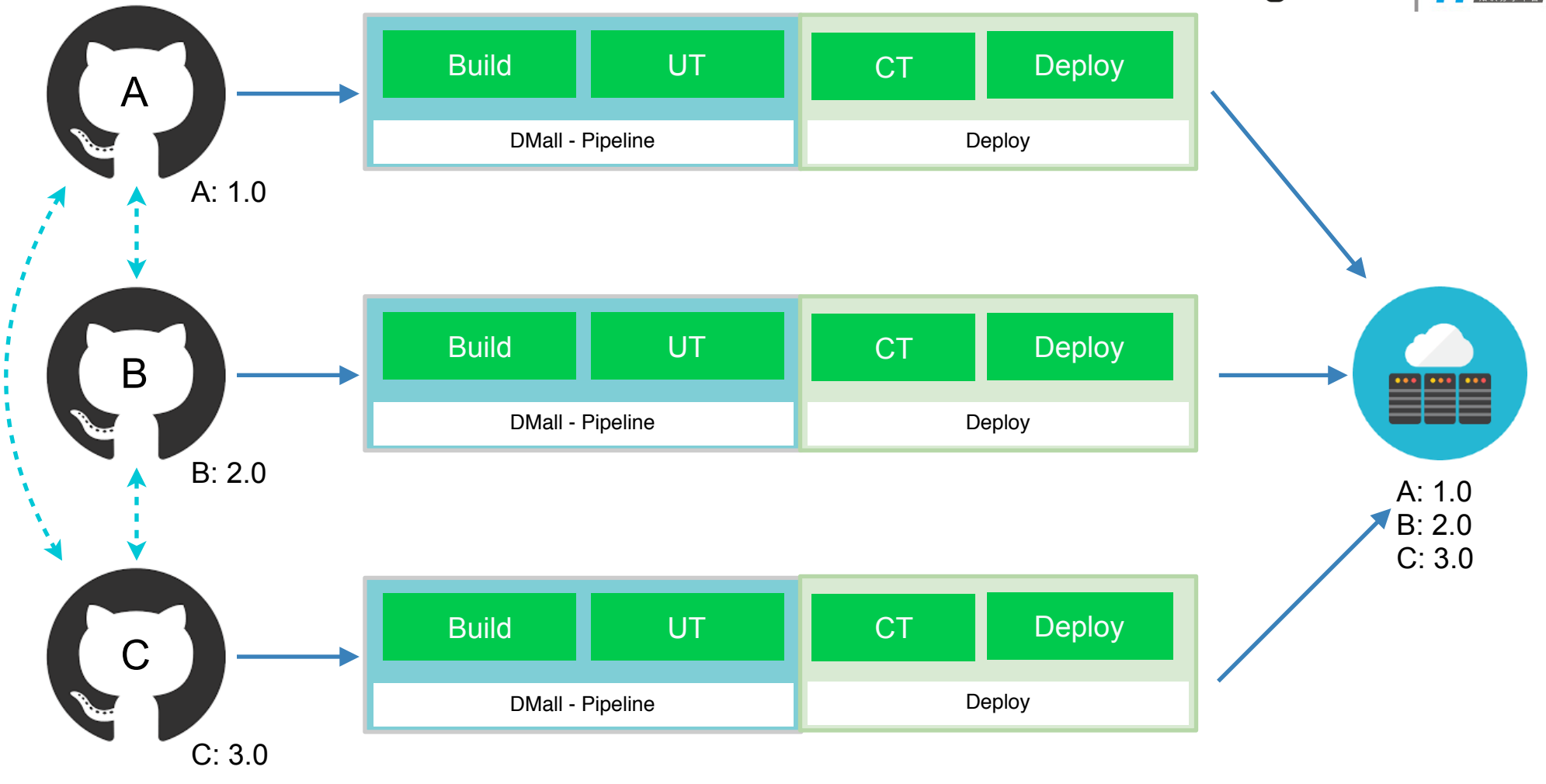

A
A: 1.1
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0

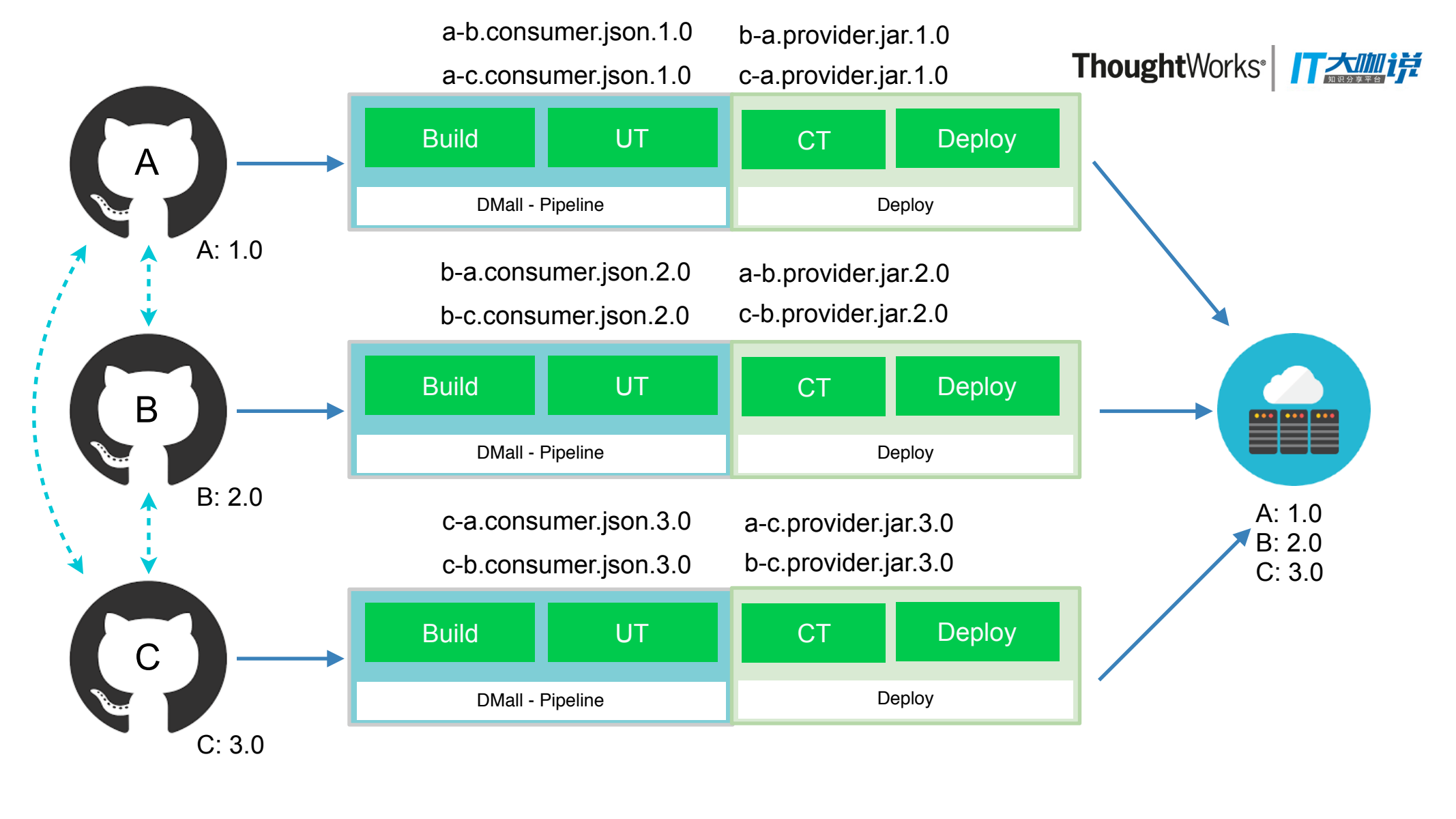

ThoughtWorks
IT大咖说
知识分享平台
a-b.provider.jar.2.0
A
A: 1.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
B
B: 2.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0

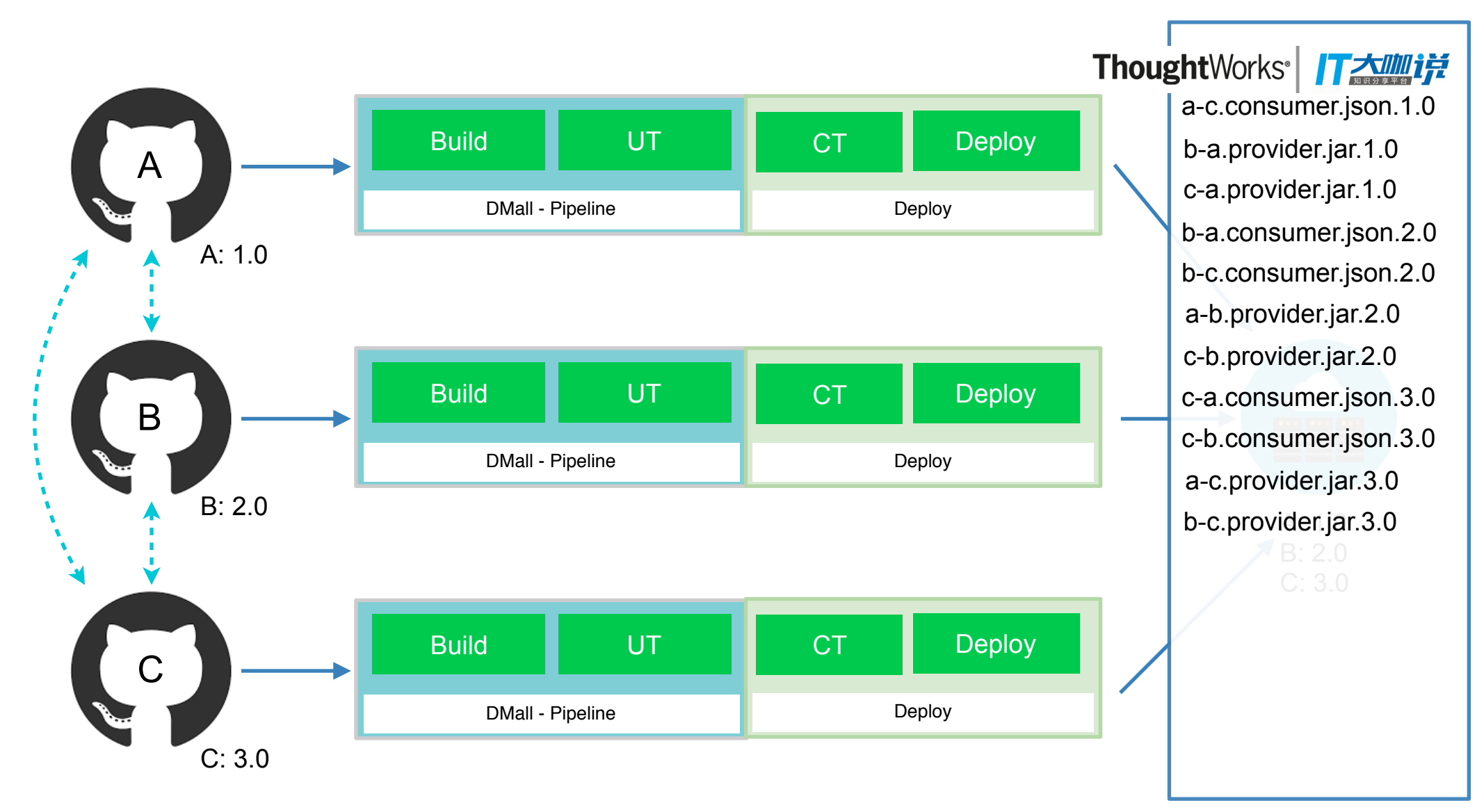

ThoughtWorks
IT大咖说
知识分享平台
a-b.provider.jar.2.0
A
A: 1.1
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-b.consumer.json.1.1
a-b.provider.jar.2.0
A: 1.0
B: 2.0
B
B: 2.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy

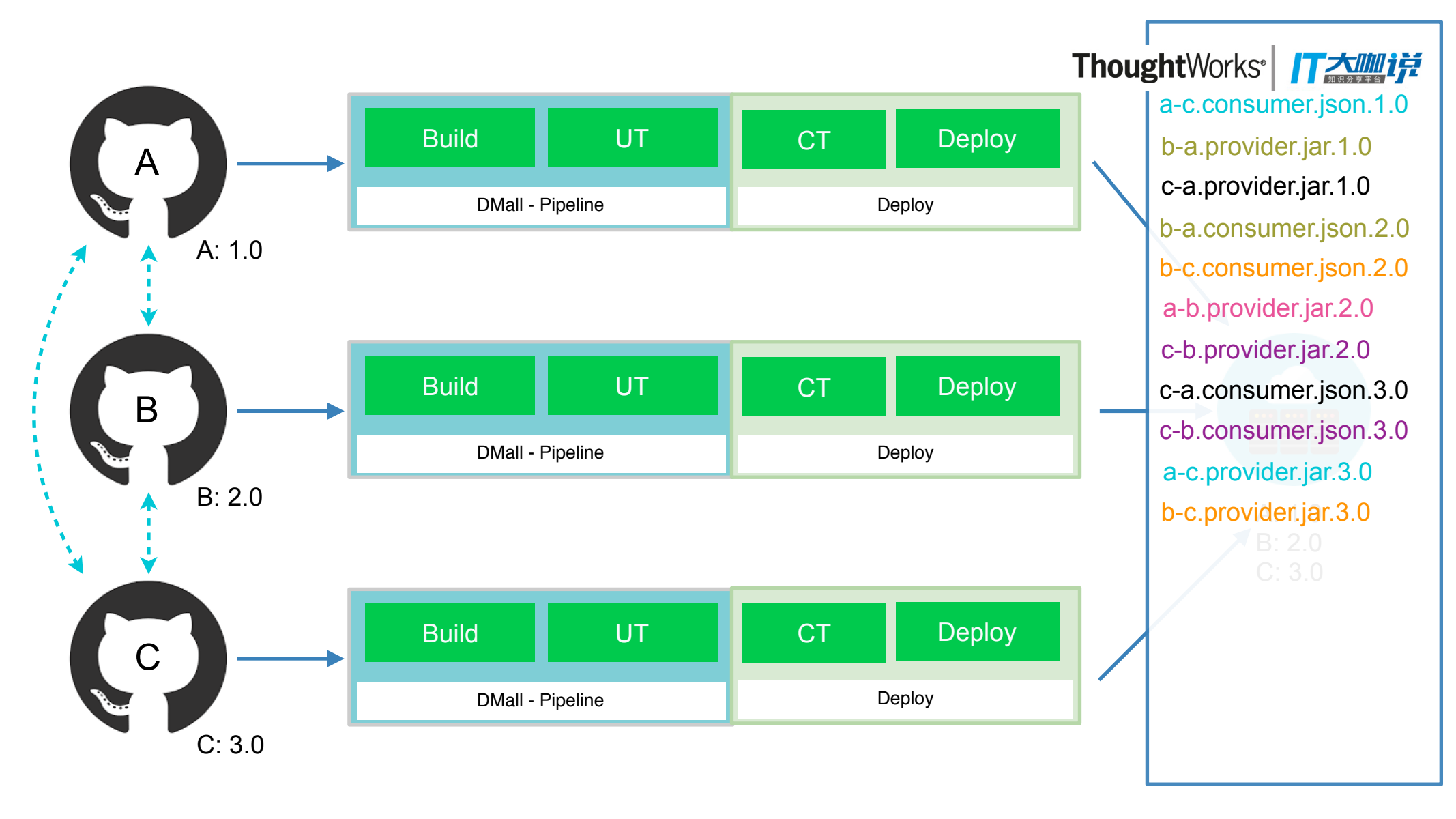

ThoughtWorks
IT大咖说
知识分享平台
a-b.provider.jar.2.0
a-b.provider.jar.2.1
A
A: 1.1
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-b.consumer.json.1.1
a-b.provider.jar.2.0
A: 1.0
B: 2.1
B
B: 2.1
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-b.consumer.json.1.0
a-b.provider.jar.2.1

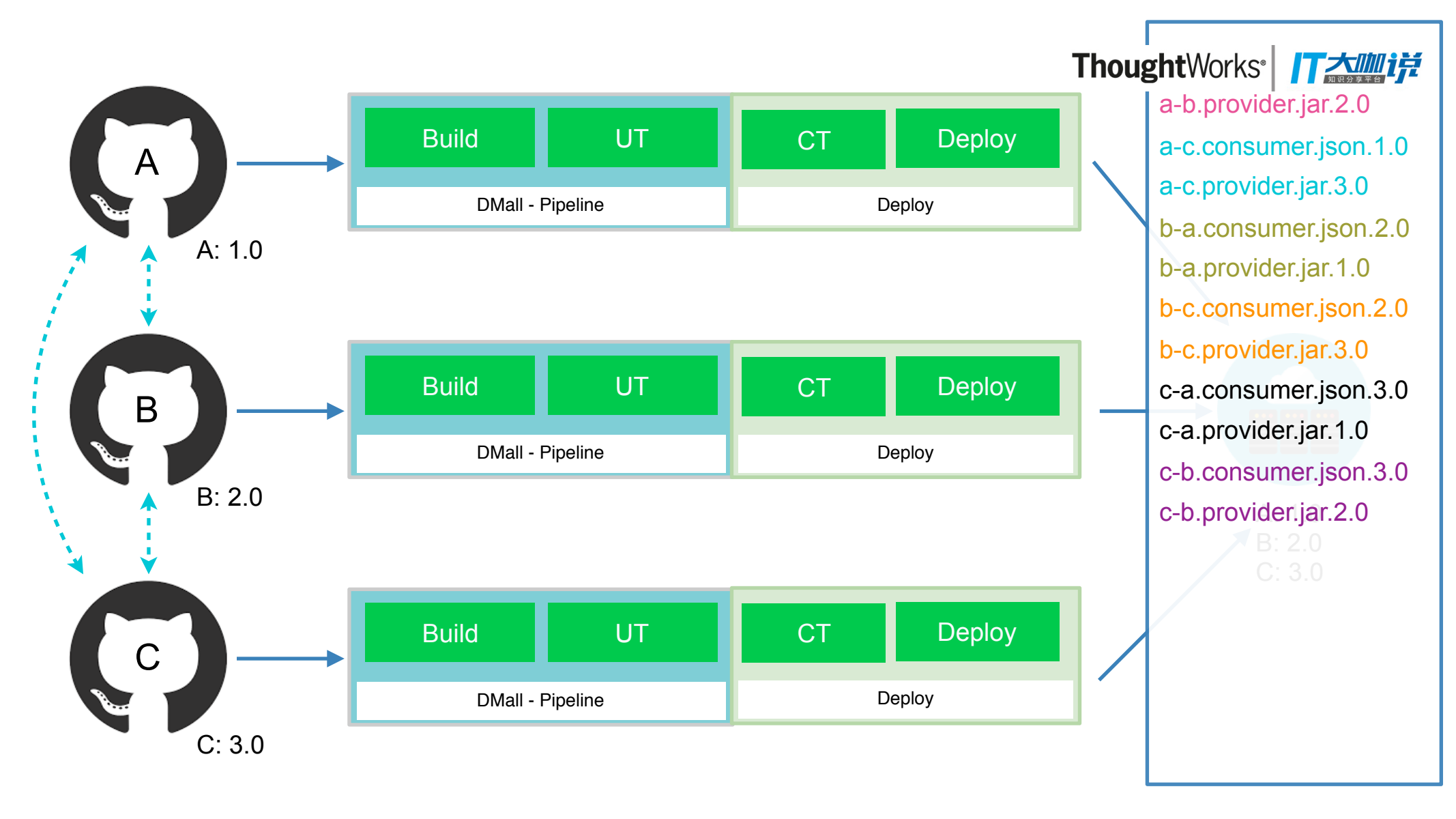

a-b.consumer.json.1.0
ThoughtWorks
IT大咖说
知识分享平台
a-b.provider.jar.2.0
a-b.provider.jar.2.1
A
A: 1.2
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-b.consumer.json.1.2
a-b.provider.jar.2.1
B
B: 2.1
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-b.consumer.json.1.0
a-b.provider.jar.2.1
A: 1.2
B: 2.1

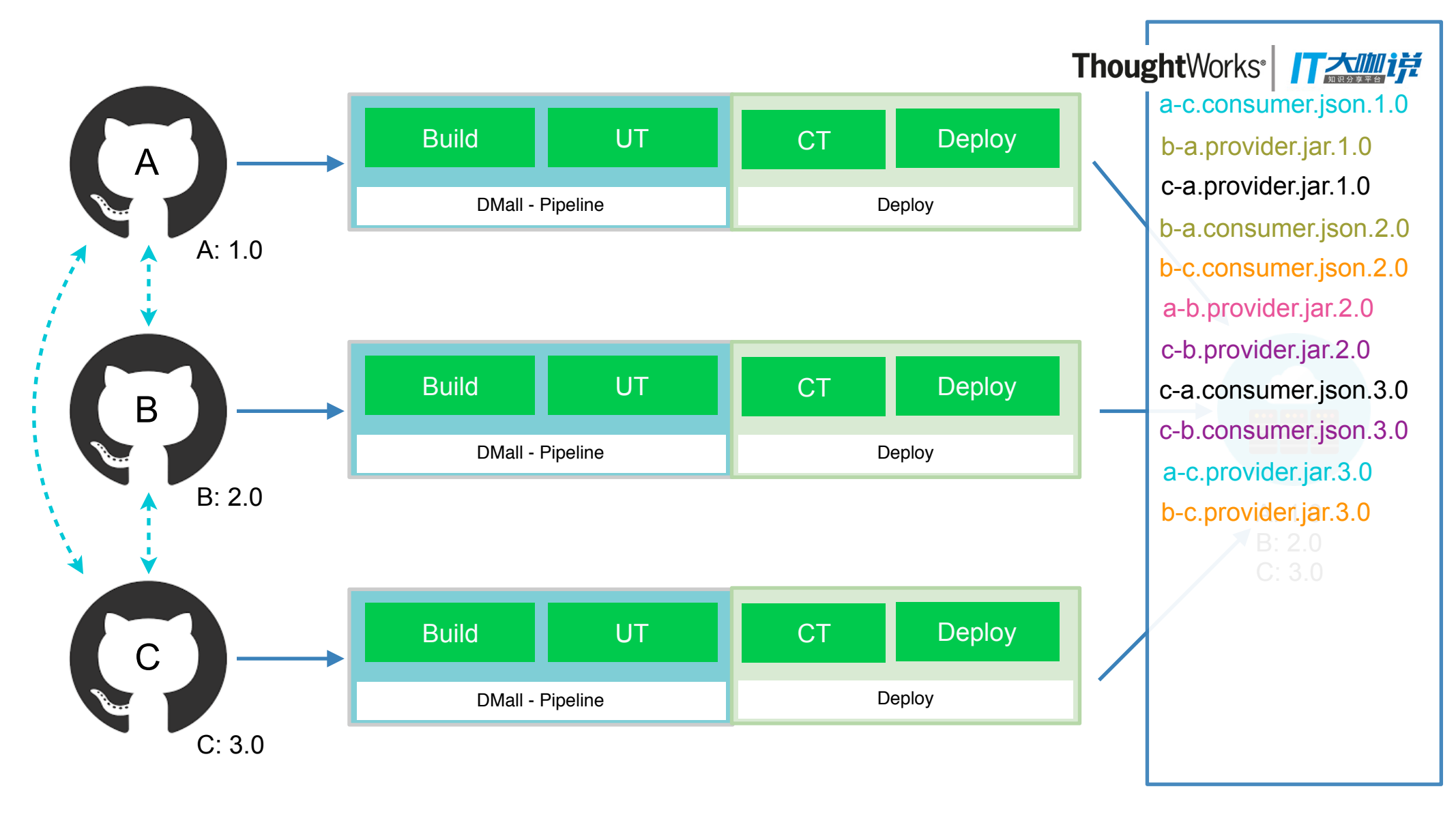

A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0

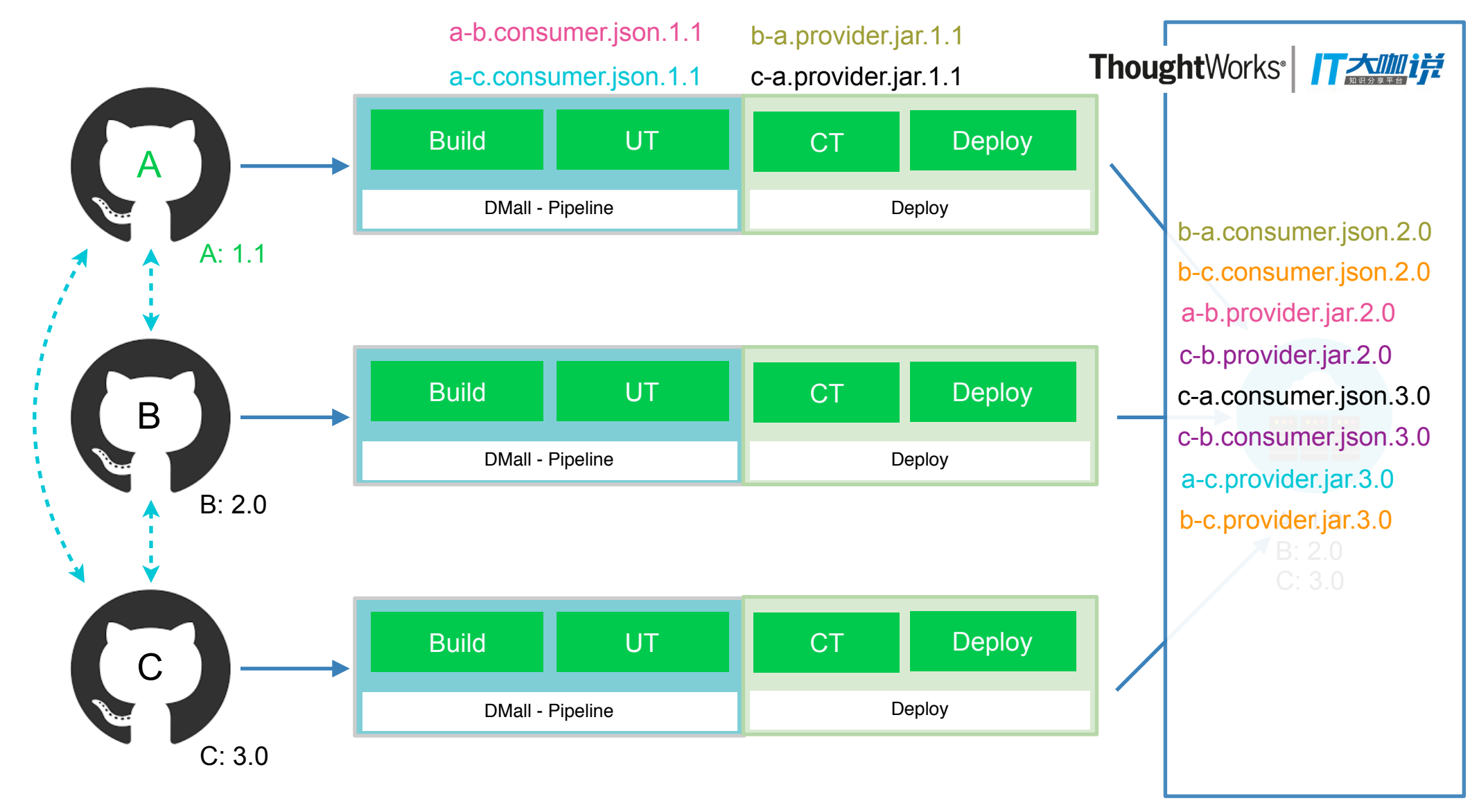

a-b.consumer.json.1.0
a-c.consumer.json.1.0
b-a.provider.jar.1.0
c-a.provider.jar.1.0
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
b-a.consumer.json.2.0
b-c.consumer.json.2.0
a-b.provider.jar.2.0
c-b.provider.jar.2.0
B
B: 2.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
c-a.consumer.json.3.0
c-b.consumer.json.3.0
a-c.provider.jar.3.0
b-c.provider.jar.3.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
A: 1.0
B: 2.0
C: 3.0

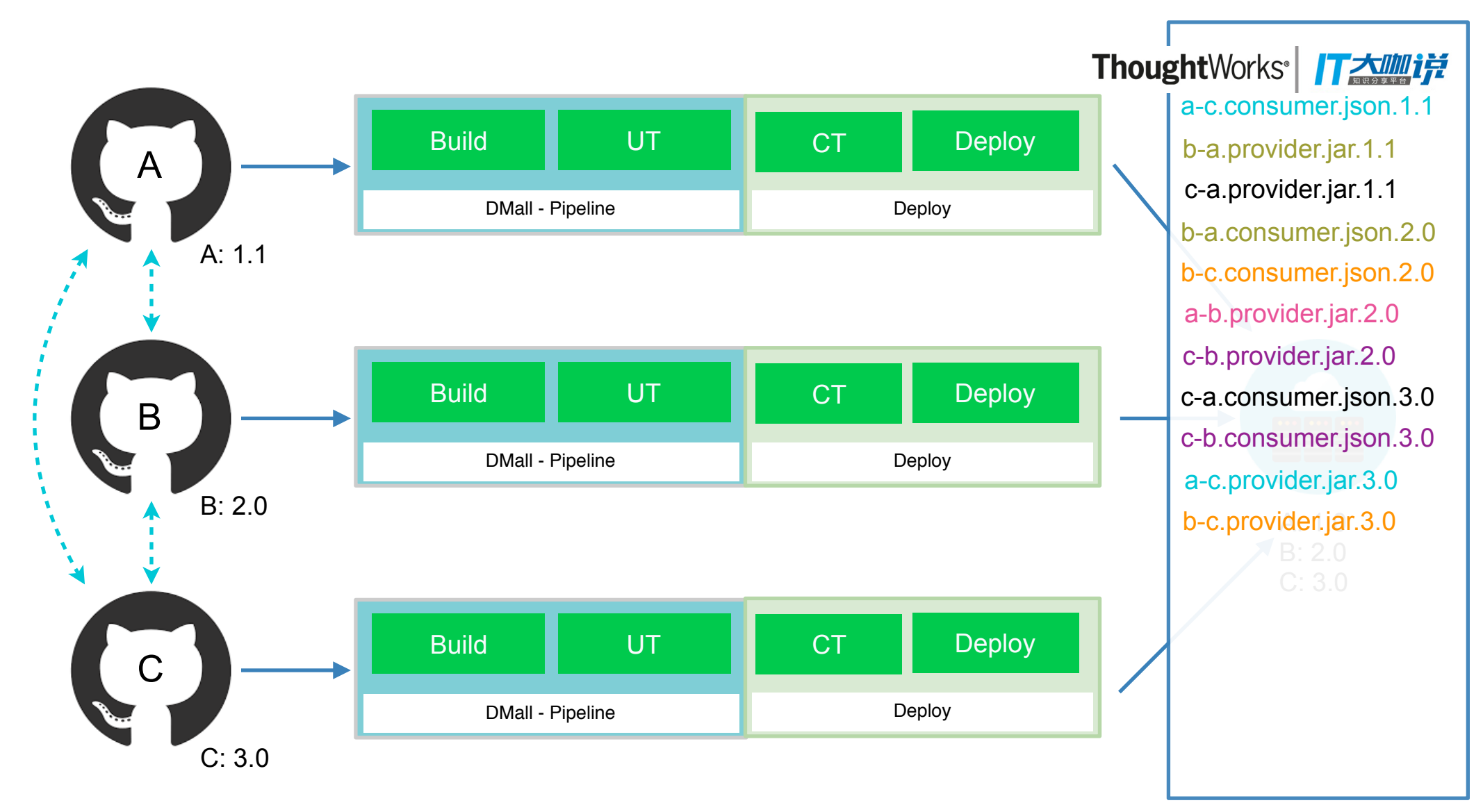

ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-c.consumer.json.1.0
b-a.provider.jar.1.0
c-a.provider.jar.1.0
b-a.consumer.json.2.0
b-c.consumer.json.2.0
a-b.provider.jar.2.0
c-b.provider.jar.2.0
c-a.consumer.json.3.0
c-b.consumer.json.3.0
a-c.provider.jar.3.0
b-c.provider.jar.3.0


ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-c.consumer.json.1.0
b-a.provider.jar.1.0
c-a.provider.jar.1.0
b-a.consumer.json.2.0
b-c.consumer.json.2.0
a-b.provider.jar.2.0
c-b.provider.jar.2.0
c-a.consumer.json.3.0
c-b.consumer.json.3.0
a-c.provider.jar.3.0
b-c.provider.jar.3.0


ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-b.provider.jar.2.0
a-c.consumer.json.1.0
a-c.provider.jar.3.0
b-a.consumer.json.2.0
b-a.provider.jar.1.0
b-c.consumer.json.2.0
b-c.provider.jar.3.0
c-a.consumer.json.3.0
c-a.provider.jar.1.0
c-b.consumer.json.3.0
c-b.provider.jar.2.0


ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.0
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-c.consumer.json.1.0
b-a.provider.jar.1.0
c-a.provider.jar.1.0
b-a.consumer.json.2.0
b-c.consumer.json.2.0
a-b.provider.jar.2.0
c-b.provider.jar.2.0
c-a.consumer.json.3.0
c-b.consumer.json.3.0
a-c.provider.jar.3.0
b-c.provider.jar.3.0


a-b.consumer.json.1.1
a-c.consumer.json.1.1
b-a.provider.jar.1.1
c-a.provider.jar.1.1
ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
b-a.consumer.json.2.0
b-c.consumer.json.2.0
a-b.provider.jar.2.0
c-b.provider.jar.2.0
c-a.consumer.json.3.0
c-b.consumer.json.3.0
a-c.provider.jar.3.0
b-c.provider.jar.3.0
B: 2.0
C: 3.0


ThoughtWorks
IT大咖说
知识分享平台
A
A: 1.1
B
B: 2.0
C
C: 3.0
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
Build
UT
CT
Deploy
DMall - Pipeline
Deploy
a-c.consumer.json.1.1
b-a.provider.jar.1.1
c-a.provider.jar.1.1
b-a.consumer.json.2.0
b-c.consumer.json.2.0
a-b.provider.jar.2.0
c-b.provider.jar.2.0
c-a.consumer.json.3.0
c-b.consumer.json.3.0
a-c.provider.jar.3.0
b-c.provider.jar.3.0

- ✓ 并发提交
- ✓ 契约变更
- ✓ 多环境支持

ThoughtWorks® | IT大咖说 知识分享平台

THANK YOU

王健

ThoughtWorks®