MiniApp URI Proposal

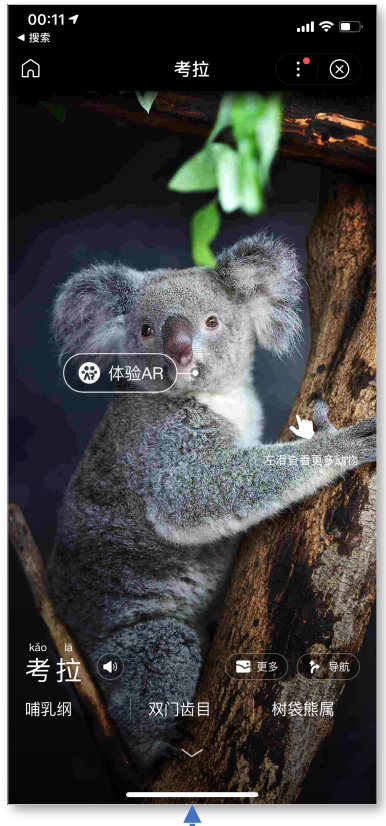
Zhou Dan

Content

- Use Case - Access a MiniApp
- Background
- MiniApp URI Syntax
- MiniApp URI Vision

Use Case - Access a MiniApp

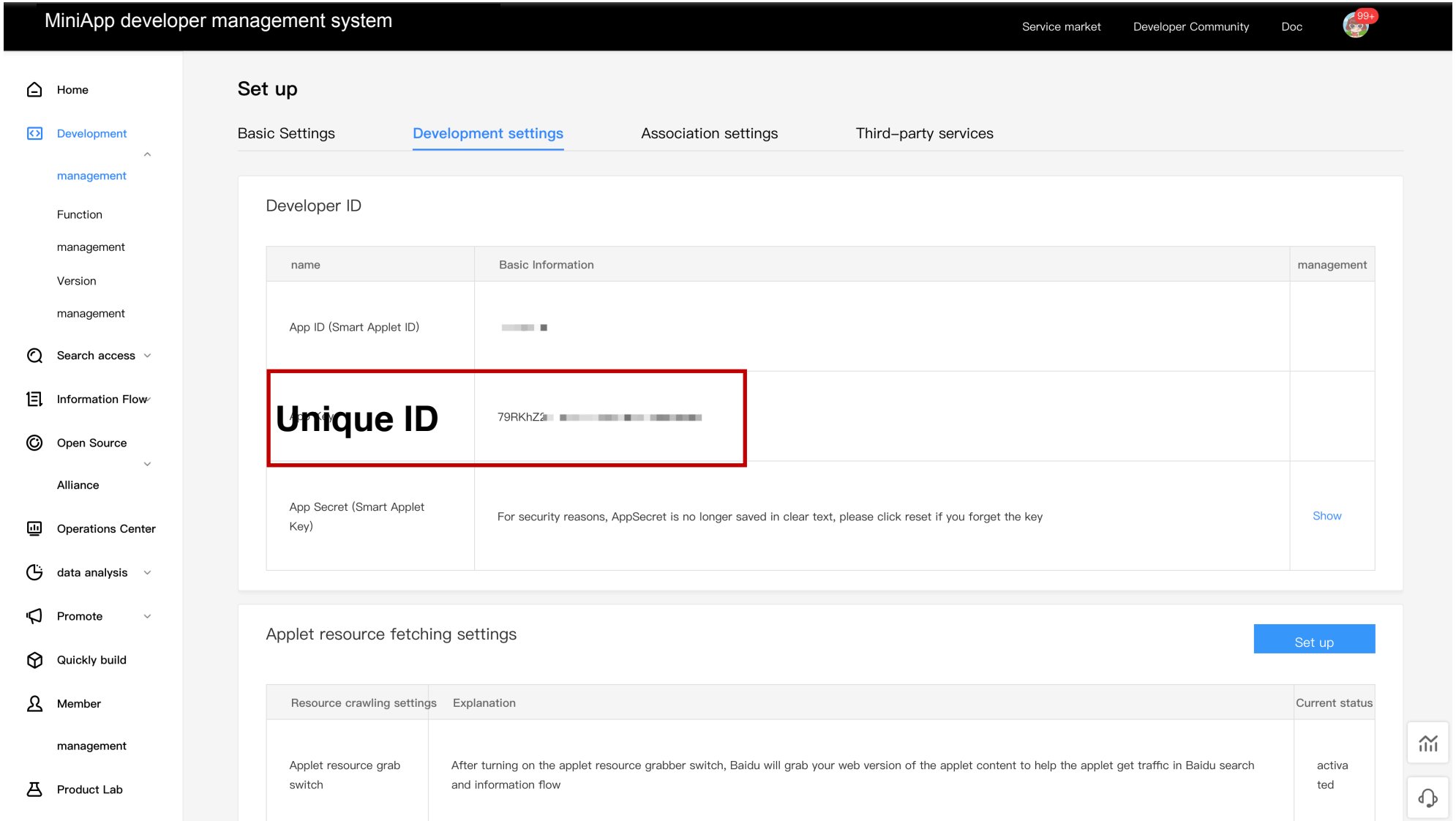
baiduboxapp://mini/123/index



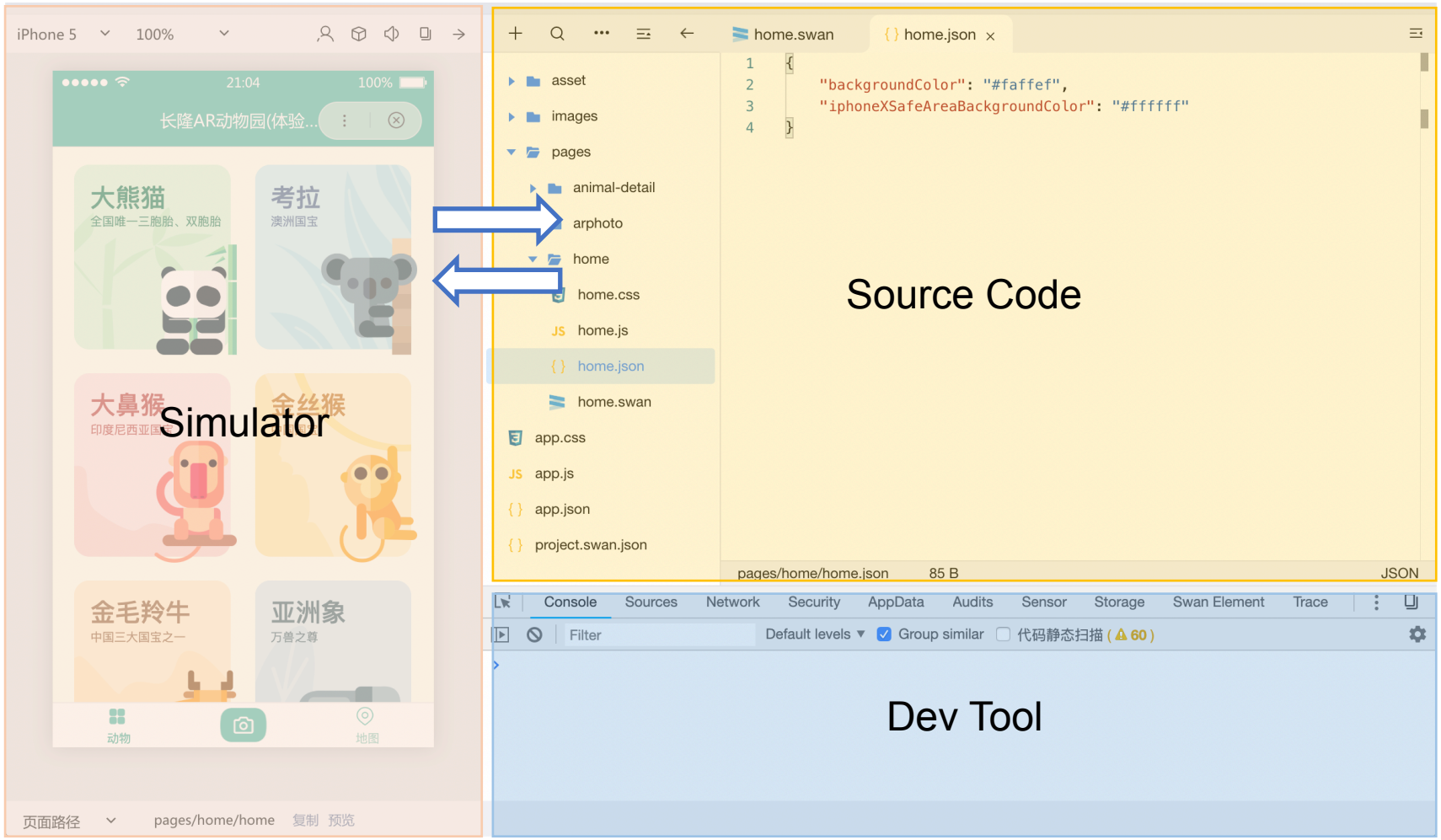
baiduboxapp://mini/123/animal?id=koala

Development Process

Register a MiniApp - Generate Unique ID

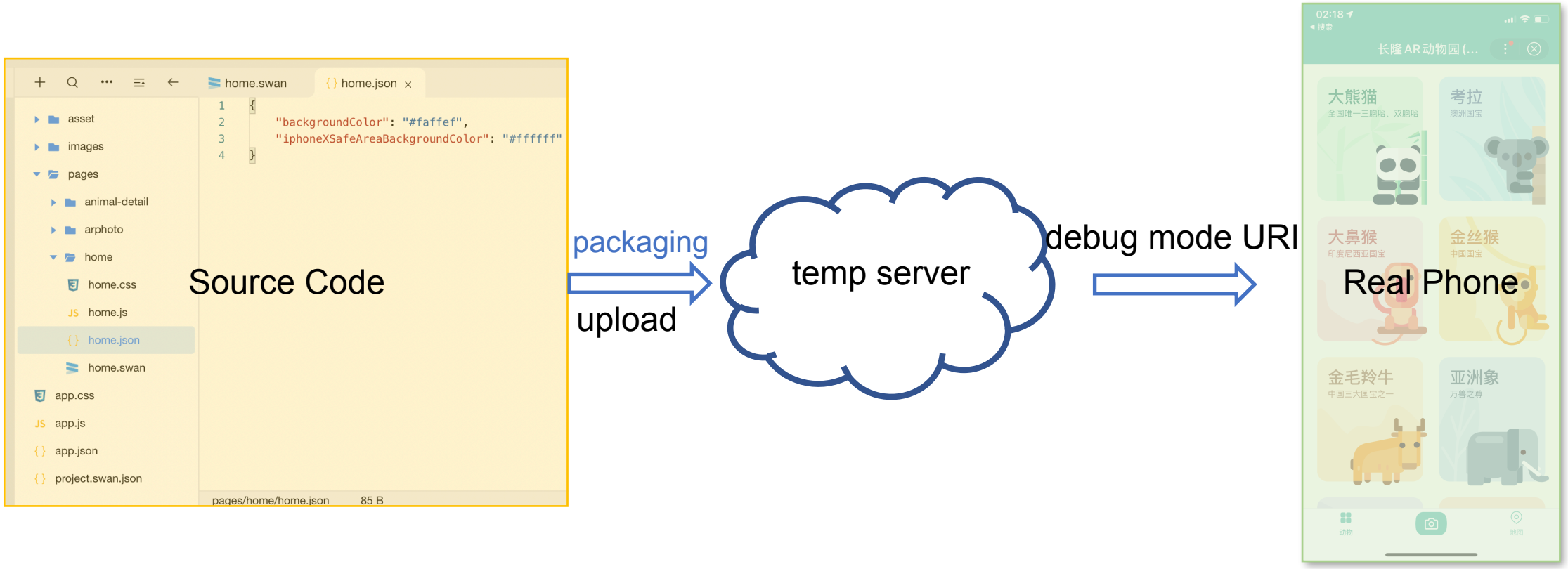


Develop & Debug MiniApp

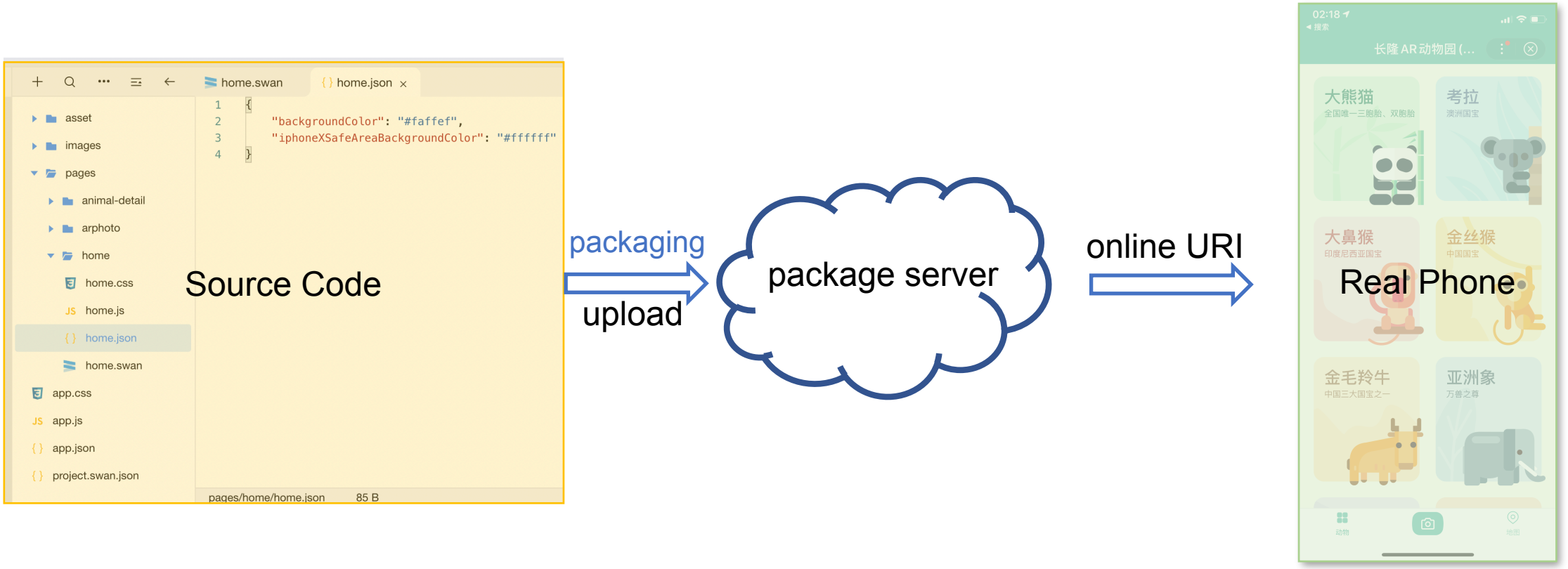


A desktop application: MiniApp IDE

Preview MiniApp - Generate Debug URI



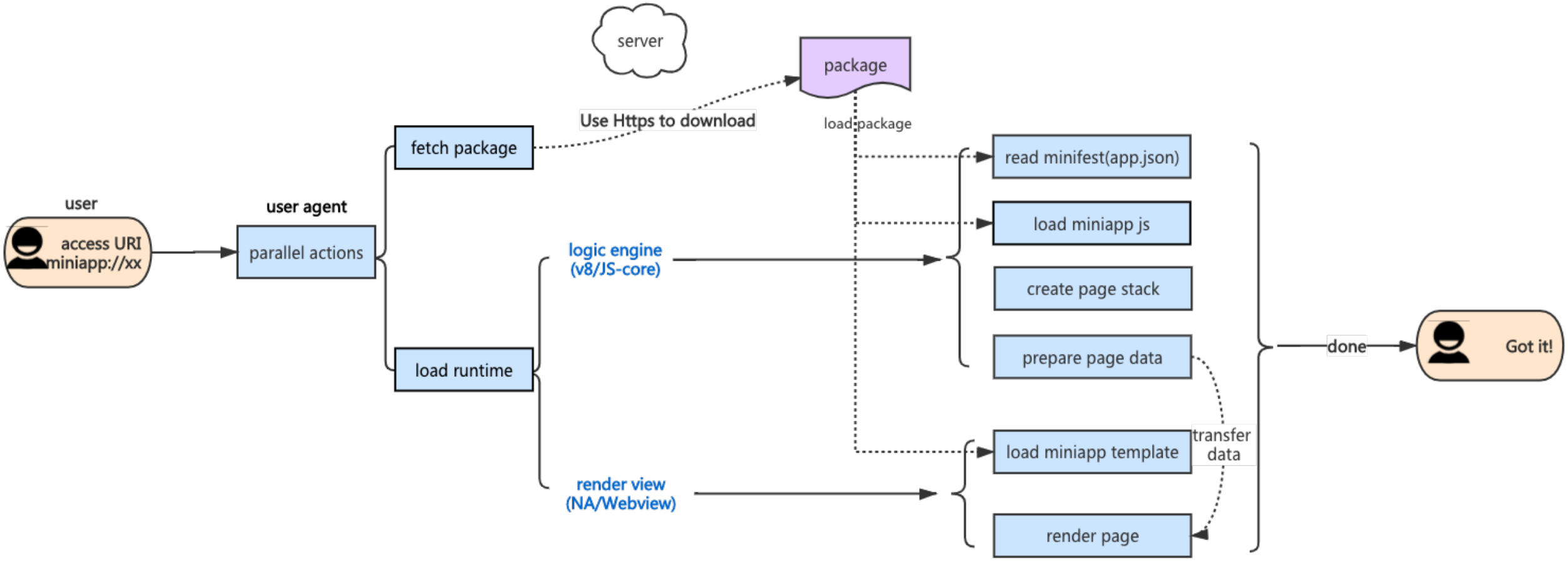
Publish MiniApp - Generate Online URI



Distribute MiniApp

- MiniApp store
- Search engine
- Smart assistant
- QR code
- Voice message
- Internet of vehicles
- Offline
- …

Access a URI



What is not in HTTP standards: **mark the resource type in URI(for loading runtime)**

Current Solution of MiniApp URI

User Agent	URI
Wechat App	weixin://dl/<path>/?appid=<appId>&businessType=<businessType>
Alipay App	alipays://platformapi/startapp?appId=<appId>&url=<path>
Quick App	http://hapjs.org/app/<appId>/[path][?key=value]
	https://hapjs.org/app/<appId>/[path][?key=value]
	hap://app/<appId>/[path][?key=value]
Baidu App	baiduboxapp://swan/<appId>/path?[key=value]
Bytedance App	sslocal://microapp?app_id=<appId>&start_page=<path>
	snssdk143://microapp?app_id=<appId>&start_page=<path>

Why We Need MiniApp URI

- MiniApp user agents are using different way to describe a MiniApp resource, which makes it difficult to access a MiniApp across platforms.
- MiniApp exist as a whole package in various environments.

--- we have to use a special rule URI to access specific resources in the MiniApp.

MiniApp URI Goals

- To define a general access protocal for
  - Navigation between MiniApps
  - Navigation between web page and MiniApp
  - Navigation between OS UI/ Native UI and MiniApp
- Identification of page inside MiniApp
  - Identify the package: id, version, package address
  - Identify the resource inside package: page path, query and fragment

URI Syntax Design Process

phase1

Q: Is it possible to use the address of downloading packages as the URI of MiniApp? Just like “https://smartapps.cn/getpkg?appKey=123&path=%2Fanimal”

A: No. The user agent need to identify whether it is a MiniApp request or not. The protocol header identification(miniapp://) will be used in pre-loading the miniapp runtime environment.

e.g., miniapp://smartapp.cn?appId=123&path=%2Fanimal

phase2

Q：As a user agent, I want more flexibility and don't want to be restricted that how and where to fetch packages

A：Yes. The name of the domain can be omitted. The only required content is the id.

e.g., miniapp://123

phase3

Q: But what if the user agent wants to specify the domain name?

A：Ok, the domain name is still a part of the authority. Like ‘miniapp://<appId>@host:port’. Yes, it looks like id replaced username.

e.g., miniapp://123@smartapp.cn

For more information, see MiniApp URI FAQ

URI Syntax Design Process

phase4

Q: Every time when the miniapp was published, it has a version number. Where is it?

A: The id is used to identify miniapps. The version number meanwhile decided which specific package was requested. It has the same status as the id number.

e.g., miniapp://foo;version=1.0.1@example.com

phase5

Q: Where should the path of miniapp be placed?

A: It makes more sense that the path of miniapp located in the path part of URI. And it will make a good separation between the query of miniapp page query.

e.g., miniapp://foo;version=1.0.1@example.com/pages/index

phase6

Q: Then how about the query of miniapp's page?

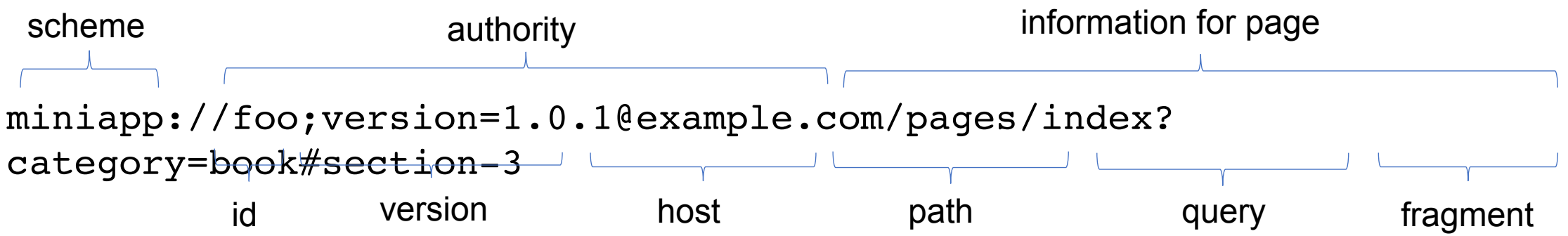
A: As mentioned above, all key info was settled down. The query of miniapp page will be placed in the query part in URI. It comes as same as the fragment part.

conclusion

miniapp://foo;version=1.0.1@example.com/pages/index?category=book#section-3

Syntax Design

```
miniappuri = scheme "://" authority path-abempty ["?" query ] ["#" fragment ]
scheme = "miniapp"
authority = id [";version=" version] ["@" host [ ":" port ]]
id = 1*unreserved
version = *unreserved
```



[For more information, see MiniApp URI Scheme proposal]

Key Scenarios

Scenario 1: A link to a MiniApp page in HTML

```
<!doctype html> <html> <a href="miniapp://foo;version=1.0.1-trial@example.com:8080/pages/index?category=book#section-3">open a MiniApp</a> </html>
```

Scenario 2: A link to a MiniApp page in JavaScript

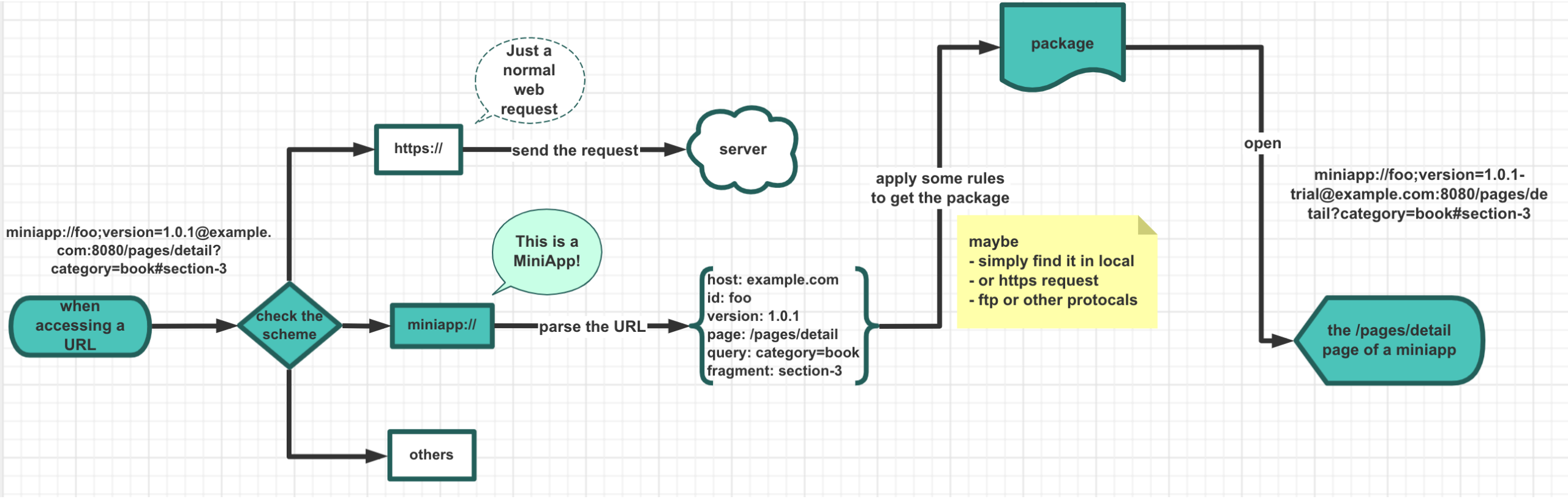
```
location.href = "miniapp://foo;version=1.0.1-trial@example.com:8080/pages/index?category=book#section-3"
```

Scenario 3: use a MiniApp URI

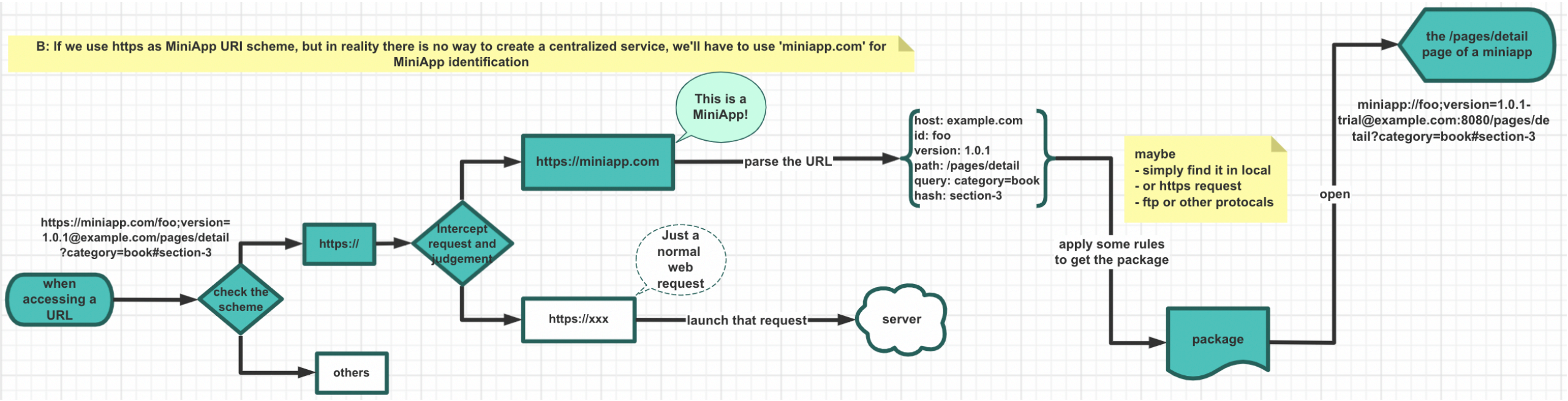
```
console.log(location.href);     // miniapp://foo;version=1.0.1-trial@example.com:8080/pages/index?k=v#bar
console.log(location.protocol); // miniapp:
console.log(location.origin);   // miniapp://foo;version=1.0.1-trial@example.com:8080
console.log(location.id);       // foo
console.log(location.version);  // 1.0.1-trial
console.log(location.host);     // example.com
console.log(location.port);     // 8080
console.log(location.pathname); // /pages/index
console.log(location.search);   // ?k=v
console.log(location.hash);     // #bar
```

Dereferencing

The dereferencing process of MiniApp URI scheme:



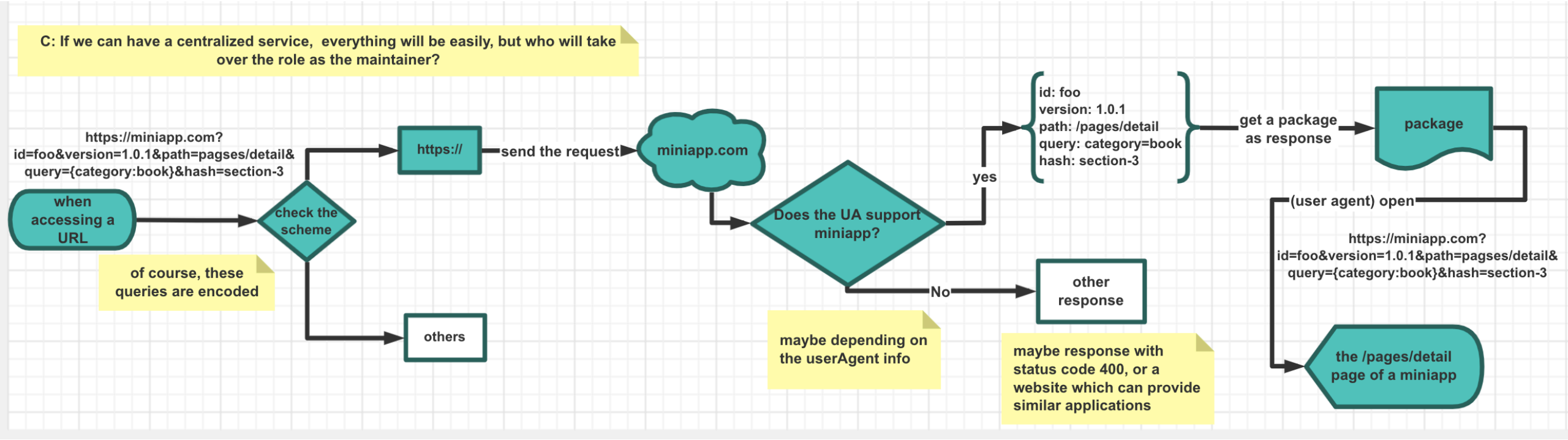
More Possibilities



Advantages:	Disadvantages:
Don’t need to create a new scheme	1. The checking process is more redundant 2. May violates the semantics of http

[For more information, see miniapp github issue #34]

More Possibilities



Advantages:	Disadvantages:
Don’t need to create a new URI syntax	1. Lack of anticipation for preparing runtime environment 2. Depends on a centralized service 3. Restrict the method to get package

Longer-Term Vision for MiniApp?

- Hear global feedback and go international
- MiniApp can run across platforms, hosts, and browsers without difference.
- MiniApp can have centralized management and ‘miniapp://id’ can uniquely identify a MiniApp in the world.

Thank you!