



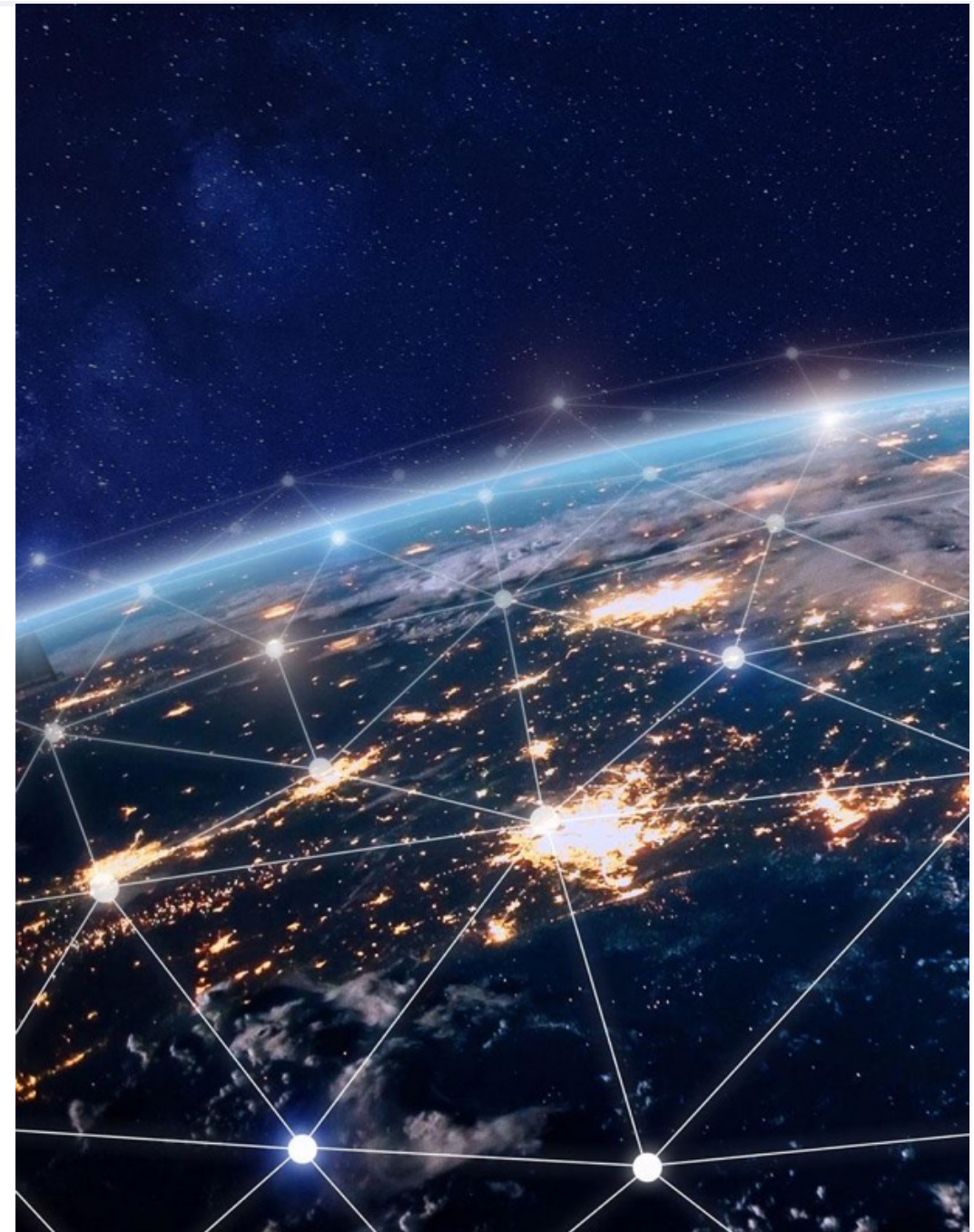
# Introduction to MiniApp (Packaging & Manifest)

**Yongjing Zhang**

**[zhangyongjing@huawei.com](mailto:zhangyongjing@huawei.com)**

**Huawei Device Co., Ltd.**

July 2020



# MiniApp Packaging

- Each MiniApp package is a specialized ZIP file (to be registered as ‘application/miniapp-pkg+zip’).
- It can be delivered to a hosting platform (user agent) through various channels, then executed by the hosting platform.

## EXAMPLE 1

```
|__manifest.json
|__app.js
|__app.css
|__pages
|   |__page1.js
|   |__page1.xml
|   |__page1.json
|   |__page1.css
|__common
|   |__componentA.js
|   |__componentA.xml
|   |__componentA.css
|   |__componentA.json
|   |__example.png
|__i18n
|   |__zh-Hans.json
|   |__en-US.json
```

- **manifest.json**: A JSON-based file responsible for the global configuration of the MiniApp.
- **app.js**: the main application logic, life-cycle management
- **app.css**: The global/default app style.
- **pages**: Contains all page-related resources, may have sub-directories.
  - **\*.js**: page scripts (app logic)
  - **\*.xml**: page layout (ffs)
  - **\*.json**: page-specific configuration (ffs)
  - **\*.css**: page-specific style
- **common**: Contains common resources (images, scripts, UI components, templates) that can be (re)used by one or multiple pages/widgets.
  - May have sub-directories. Details are ffs (e.g. components)
- **i18n**: Contains multi-language configuration files, each stands for a specific language.
  - The JSON file format is to be specified.

**Security considerations:** A MiniApp package should be protected by digital signatures (e.g. PKCS#7) of an author and one or more distributors to ensure its integrity and trustworthiness. Details are ffs.

# Processing the MiniApp Package\*

- Step 1 - Acquire a MiniApp Archive (.ma\*)
  - The package delivering channel can be flexible, e.g. web downloading, app store, offline.
- Step 2 - Verify the MiniApp Package File (digital signatures)
  - To ensure the package comes from a trusted author/distributor, and the package is not tampered..
- Step 3 - Locate & Process the Manifest File (manifest.json)
  - Check platform compatibility and permissions, get default UI style, page list, etc.
- Step 4 - Prepare the Runtime
  - Logic layer and view layer
- Step 5 - Locate & Process the Start Files (app.js, first page)
  - Find the entrance and start the MiniApp life cycle.

*\*Note: The whole process is currently under discussion.*

# MiniApp Packaging vs. other Packaging Technologies

Technology	Client Platform	Content	Web Dependency	Format	Digital Signature	Remarks
MiniApp	Super-App, OS, Browser*	Application-resources (e.g. app logics, page layouts, UI components)	No (may or may not use web resources)	ZIP-based	PKCS#7*	More like a mobile app packaging (e.g. APK)
WPACK	Browser	Web content (HTTP exchanges)	Yes (a web origin is always assumed, even offline)	CBOR-based	HTTP header extension	HTTP-specific design (archiving format, digital signature) is not suitable for MiniApp.
WebPackage (obsolete)	Browser	Web content	Yes (a web origin is always assumed, even offline)	Streamable Package Format (.pack)	n/a	Similar as above.
Widgets (obsolete)	Browser, OS	Widget files (html start files, XML configuration, icons ...)	No (may or may not use web resources)	ZIP-based	XMLDSIG-based	XML-based configuration and digital signature doesn't fit MiniApp.
LPF	e-Readers	Digital publications, configuration	No	ZIP-based	unknown*	Publication-specific configuration doesn't fit MiniApp. Lack of security protection

Although MiniApp leverages popular Web technologies like JS, CSS, it's more like a native mobile app rather than a snapshot of a web-content collection.

Note: \* denotes the feature that is under discussion/development.

# MiniApp Manifest

The MiniApp Manifest is a **JSON-based** file that enables developers to set up basic information, window style, page route and other global configuration of a MiniApp.

- Basic metadata about the MiniApp

- Version control of the MiniApp, to ensure the platform compatibility.

- Configuration of page routing and window style

- Permission declaration for local resource/API access

Attribute	Type	Required	Description
dir	string	No	Text direction
lang	string	No	Language tag
appId	string	Yes	ID of the MiniApp
appName	string	Yes	App name
shortName	string	No	Short name
description	string	No	Description
icons	Array	Yes	Application icons
versionName	string	Yes	Version name
versionCode	number	Yes	Version code
minPlatformVersion	string	Yes	Minimum platform version supported
pages	Array	Yes	Route information
window	Object	No	Window style
widgets	Array	No	Widget
reqPermissions	Array	No	Required permissions

## EXAMPLE 1

```
{
  "dir": "ltr",
  "lang": "en-US",
  "appId": "org.w3c.miniapp",
  "appName": "MiniApp Demo",
  "shortName": "MiniApp",
  "versionName": "1.0.0",
  "versionCode": 1,
  "description": "A Simple MiniApp Demo",
  "icons": [
    {
      "src": "common/icons/icon.png",
      "sizes": "48x48"
    }
  ],
  "minPlatformVersion": "1.0.0",
  "pages": [
    "pages/index/index",
    "pages/detail/detail"
  ],
  "window": {
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "Demo",
    "navigationBarBackgroundColor": "#f8f8f8",
    "backgroundColor": "#ffffff",
    "fullscreen": false
  },
  "widgets": [
    {
      "name": "widget",
      "path": "widgets/index/index",
      "minPlatformVersion": "1.0.0"
    }
  ],
  "reqPermissions": [
    {
      "name": "system.permission.LOCATION",
      "reason": "To show user's position on the map"
    },
    {
      "name": "system.permission.CAMERA",
      "reason": "To scan the QR code"
    }
  ]
}
```

# MiniApp Manifest vs. Web App Manifest

Aspects	MiniApp Manifest	Web App Manifest	Remarks
Basic info	dir, lang, appName (name), shortName (short_name), description, icons		Both contains a common set of basic metadata
Extras	tbd.	theme_color, iarc_rating_id, categories, screenshots, shortcuts...	MiniApp may incorporate other useful attributes from Web App Manifest in future development.
Platform	appId, versionName, versionCode, minPlatformVersion	n/a	MiniApp requires a strong means of <b>application version management</b> to ensure the compatibility with the hosting platform comparing to the web environment
Permission	reqPermissions	n/a	MiniApp requires <b>strict permission control</b> since the hosing platform may expose more sensitive data and functions comparing to the web browser.
Application components	window	Partial support (e.g. background_color, orientation, fullscreen)	MiniApp pages/widgets are organically composed <b>views/activities</b> of the MiniApp rather than independent web pages, so need to be <b>configured properly into a common look and feel</b> (e.g. navigation bar, scrolling behavior, width adaptation)
	pages, widgets	n/a	

As MiniApp has different assumption on the **hosting platform** and **different formation of the application components**, the manifest needs to provide specific means for the platform compatibility, permission control, and UI consistency.