



Steering Committee

February 9th, 2022

Agenda

- News in brief
- Technology
- Community
- Workgroups

News in brief



01 Portal update

- New “Join In” section highlights project proposals
- Revised “About” section
- Zach working on a “how to contribute” guide

02 Triple event 29-30 June, Paris

- IOT World Paris
 - OW2 - 15 year anniversary
 - QAI - 1 year anniversary
- Stand? }
Party time! } Quick App demo opportunity

03 MWC 30/03, Barcelona

- Mobile World Congress
- ½ day QAI event being organised



EVENTS RADAR

H1 2022

MARCH

- MWC (02nd)



APRIL

- Systematic Hub Day (tbc)
- The WebConference (25-29)



MAY

- ?



JUNE

- OW2con'2022 (8-9th)
- Anniversary party
- IOT World



JULY

- ?



Big events



BARCELONA 02/03

½ day event
Talks + Interactive



ONLINE 8-9 JUNE

1hr slot
3 talks + 1 round table
CFP deadline 27/02

OW2 + QAI
anniversary

29-30 JUNE

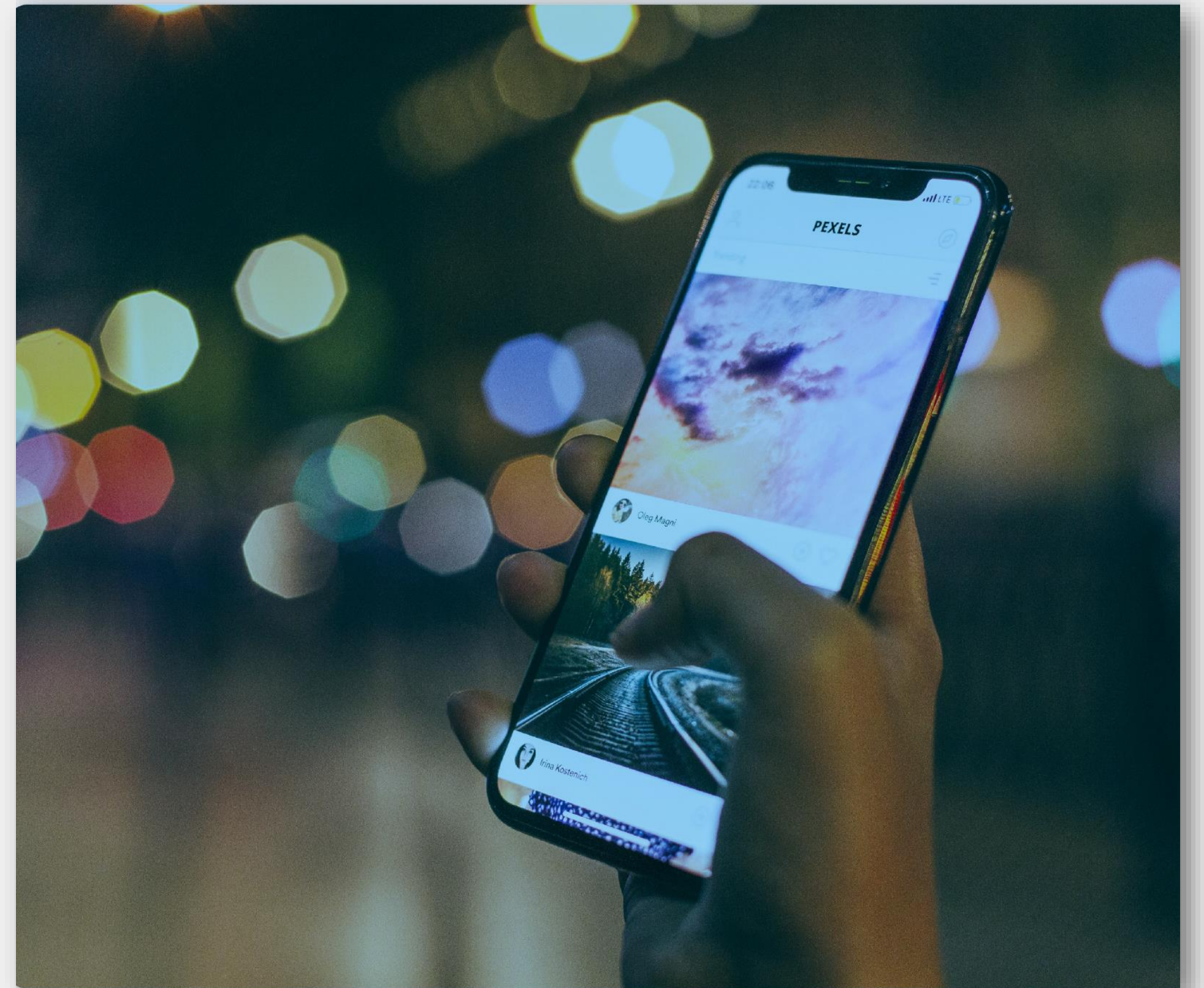
Quick app “corner” with demos?

- For example
- The business of Quick Apps
 - Why mobile needs a change
 - Web 3.0 (Quick Apps, Solid, ...)

...

Technology

Standards, Tools, Documents, Code



W3C standards

Specifications

- Manifest
 - Packaging
 - Widgets
 - Lifecycle
 - Addressing
-
- White Paper
 - MiniApp for IoT
 - Components

CONTENTS

Abstract

Status of This Document

Introduction

Problems

What is MiniApp?

Can we just use PWA?

Use cases

Case 1: Shared Bicycle Service

Case 2: AR Zoo

Case 3: MiniApp for IoT

MiniApp Overview

Core Features

Separate the View Layer from the Logic Layer

Rich APIs and Components

MiniApp Constructor

MiniApp Widgets

Single-Instance, Multi-Entries

Performance and User Experience

MiniApp Market

Working with the Web

Application Lifecycle

Hybrid Rendering

Transition Animation

Standardise the Package Constructor of MiniApp

Standardise the navigation to a MiniApp page

MiniApp Widgets

Performance and Tuning

Define an event of time to interactive in MiniApp

Graphics and Media

3D Model Element

Face Tracking

Hand gestures tracking and recognition

Low level AR APIs based on ARCore and ARKit

Security and Privacy Consideration

Way Forward in W3C

Glossary

Gap Analysis

Acknowledgments

MiniApp Standardization White Paper

W3C Editor's Draft 21 January 2022



More details about this document

This version:

<https://w3c.github.io/miniapp/white-paper/>

Latest published version:

<https://www.w3.org/TR/mini-app-white-paper/>

Latest editor's draft:

<https://w3c.github.io/miniapp/white-paper/>

History:

<https://www.w3.org/standards/history/mini-app-white-paper>

[Commit history](#)

Editors:

Anqi Li ([W3C](#))

Qing An ([Alibaba](#))

Dan Zhou ([Baidu, Inc](#))

Wanming Lin ([Intel Corporation](#))

Kaining Yuan ([Intel Corporation](#))

Yinli Chen ([Xiaomi](#))

Xiaowei Jiang ([Xiaomi](#))

Fuqiao Xue ([W3C](#))

Former editors:

Dapeng Liu ([Alibaba](#))

Hongru Zhu ([Alibaba](#))

Qingqian Tao ([Baidu, Inc](#))

Zhixing Lei ([Baidu, Inc](#))

Zhiqiang Yu ([Huawei](#))

Lei Zhao ([China Mobile](#))

Feedback:

[GitHub w3c/miniapp](#) (pull requests, new issue, open issues)

public-miniapps-wg@w3.org with subject line [mini-app-white-paper] - message topic - (archives)

Copyright © 2019-2022 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.

§ Abstract

This document introduces a new format for mobile application, named MiniApp, which is a very popular hybrid solution relying on Web technologies but also integrates with capabilities of Native Apps.

§ Status of This Document

This section describes the status of this document at the time of its publication. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <https://www.w3.org/TR/>.

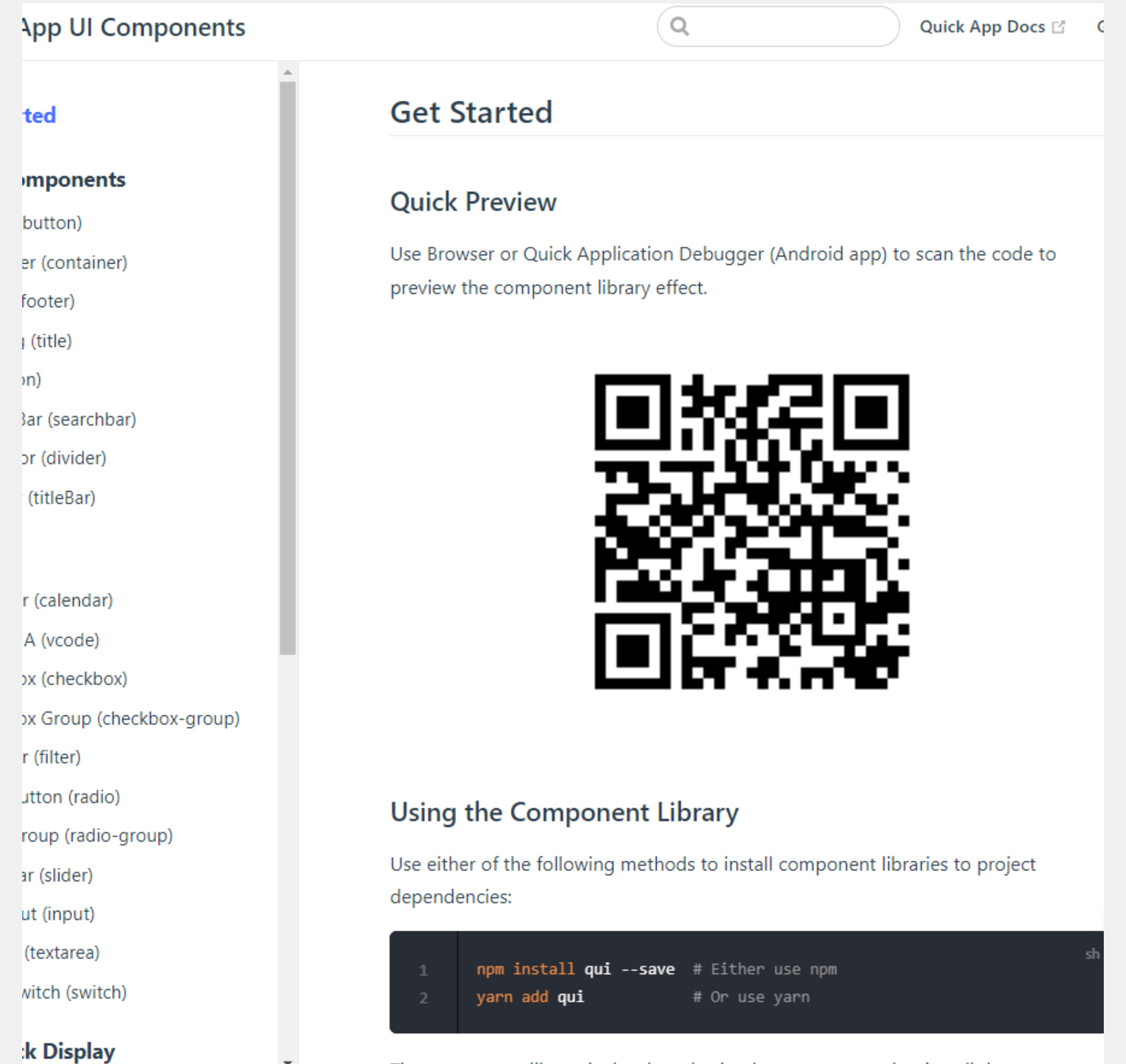
This document was published by the [MiniApps Working Group](#) as an Editor's Draft.

Tools

UI Components for quick apps

- <https://espinr.github.io/quick-app-ui/>
- Translation from quickapp.cn
- Still to refine

- Web server → emulation on browser



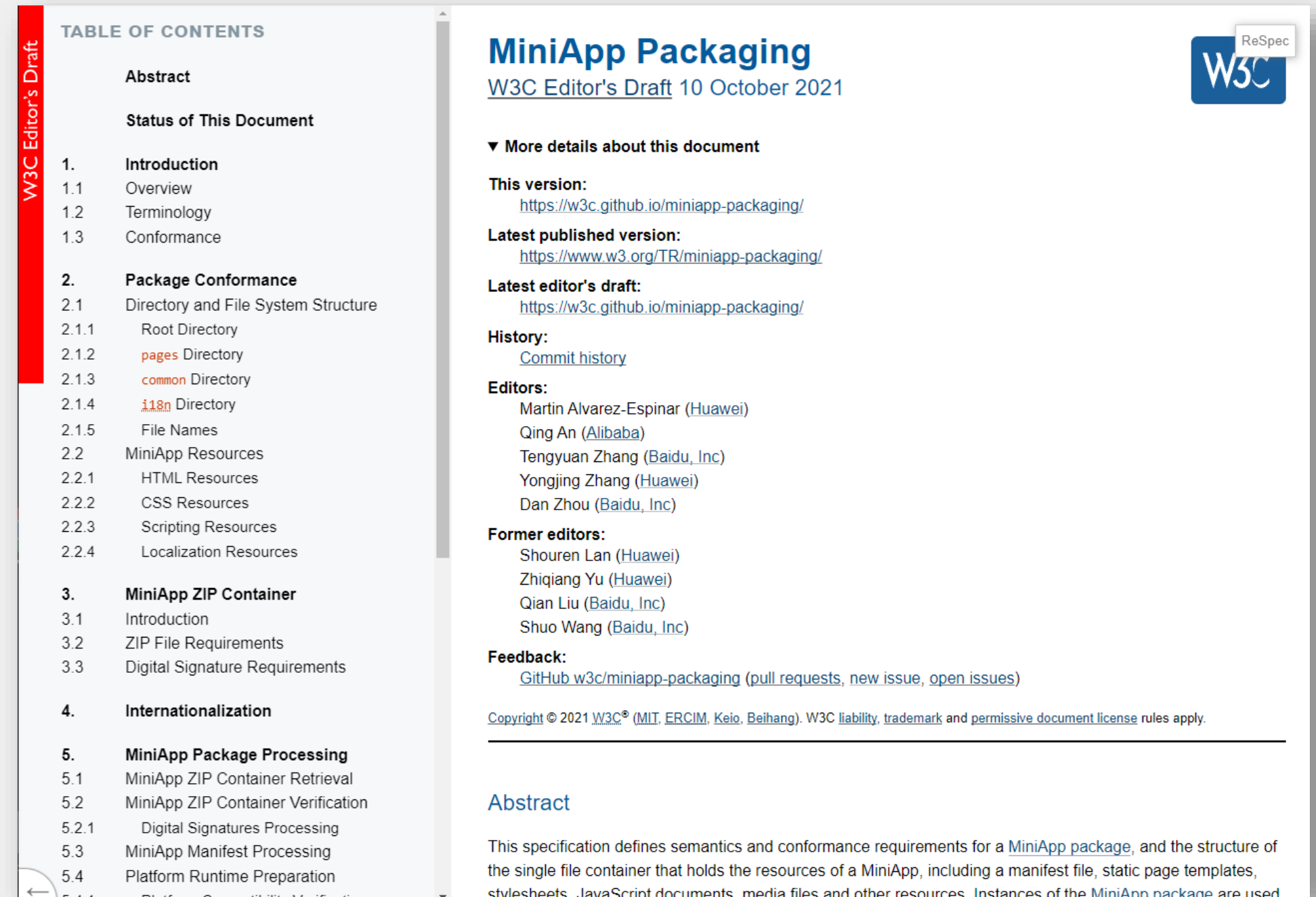
The screenshot shows the 'App UI Components' website. On the left is a navigation menu with a search bar and a list of components: button, container, footer, title, searchbar, divider, titleBar, calendar, vcode, checkbox, checkbox-group, filter, radio, radio-group, slider, input, textarea, and switch. The main content area is titled 'Get Started' and includes a 'Quick Preview' section with a QR code and instructions to use a browser or debugger. Below that is a 'Using the Component Library' section with terminal commands for installing the library using npm or yarn.

```
1 npm install qui --save # Either use npm
2 yarn add qui # Or use yarn
```


Packaging

Defines the **logical** (pages, components, styles...) and **physical structure** (file system directories, signatures...) of a **MiniApp**.

- Spec: <https://w3c.github.io/miniapp-packaging/>
- Repo: <https://github.com/w3c/miniapp-packaging>
- Status: **Ready for FPWD**



The screenshot shows the 'MiniApp Packaging' document page, which is a W3C Editor's Draft. The page features a 'TABLE OF CONTENTS' on the left and a main content area on the right. The 'TABLE OF CONTENTS' includes sections for Abstract, Status of This Document, Introduction (with sub-sections for Overview, Terminology, and Conformance), Package Conformance (with sub-sections for Directory and File System Structure, Root Directory, pages Directory, common Directory, and i18n Directory), File Names, MiniApp Resources (with sub-sections for HTML, CSS, Scripting, and Localization Resources), MiniApp ZIP Container (with sub-sections for Introduction, ZIP File Requirements, and Digital Signature Requirements), Internationalization, and MiniApp Package Processing (with sub-sections for MiniApp ZIP Container Retrieval, MiniApp ZIP Container Verification, Digital Signatures Processing, MiniApp Manifest Processing, and Platform Runtime Preparation). The main content area includes the title 'MiniApp Packaging', the subtitle 'W3C Editor's Draft 10 October 2021', a 'ReSpec' logo, and a 'More details about this document' section. This section provides links for 'This version', 'Latest published version', and 'Latest editor's draft'. It also lists the 'History' (Commit history), 'Editors' (Martin Alvarez-Espinar, Qing An, Tengyuan Zhang, Yongjing Zhang, Dan Zhou), and 'Former editors' (Shouren Lan, Zhiqiang Yu, Qian Liu, Shuo Wang). A 'Feedback' section links to GitHub pull requests, new issues, and open issues. A copyright notice at the bottom states 'Copyright © 2021 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and permissive document license rules apply.' The 'Abstract' section begins with 'This specification defines semantics and conformance requirements for a MiniApp package, and the structure of the single file container that holds the resources of a MiniApp, including a manifest file, static page templates, stylesheets, JavaScript documents, media files and other resources. Instances of the MiniApp package are used'.

MiniApp Packaging

- How to **create and handle MiniApp packages**
- Definition of the **main terms** of the MiniApps:
 - MiniApp; page, resources, widget...
- Requirements of the **file system** (directories, filenames, etc.) →
- Main resources:
 - *.html, .css, .js, manifest.json*
 - **Localization (i18n)** resources
- **MiniApp ZIP container:**
 - Logic and physical requirements
 - Signature mechanisms

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

Types of Resources

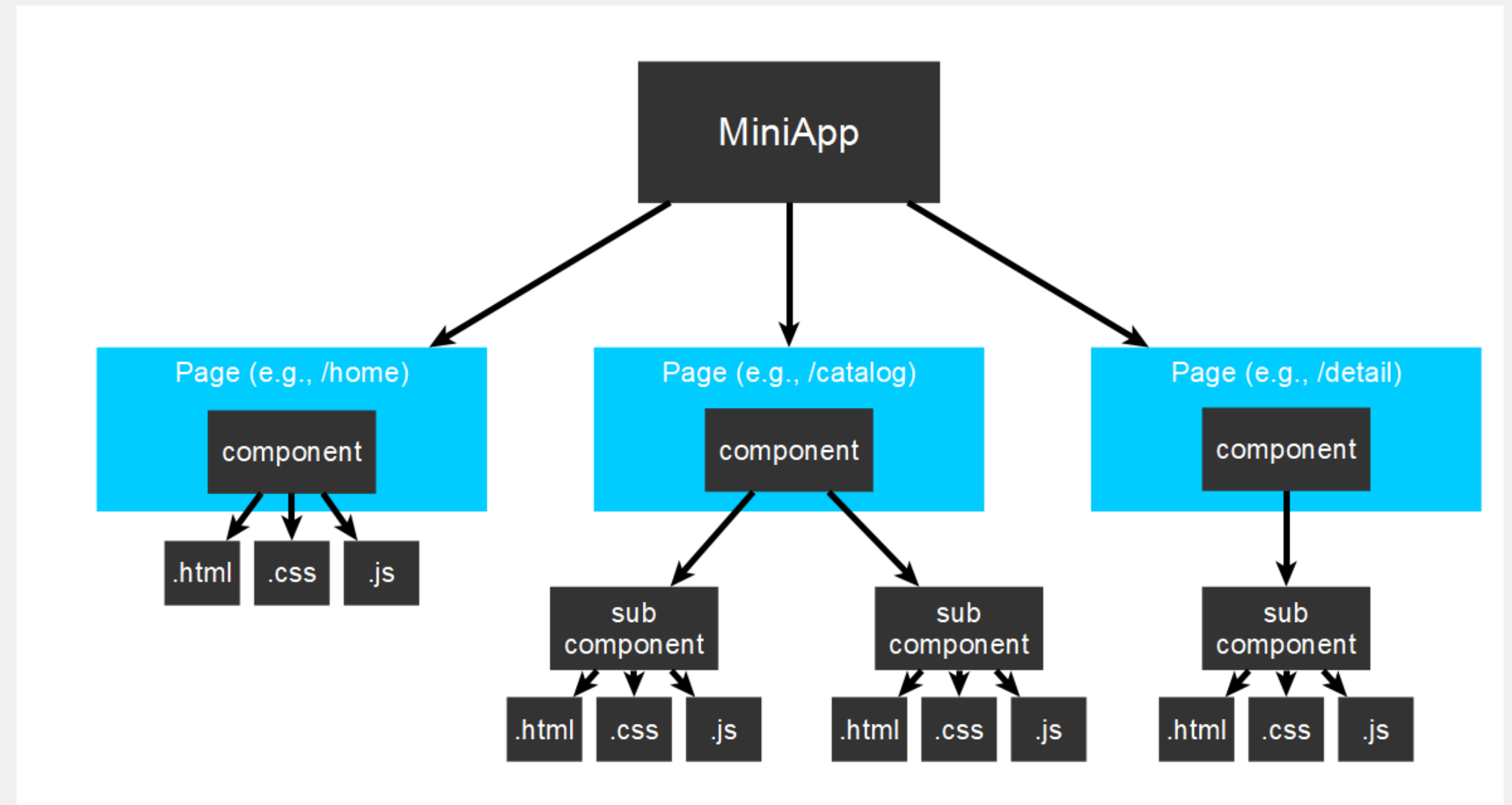
HTML → HTML profile?

- **HTML elements** may be supported as they are
- HTML **Web components** widely used in MiniApps
- To create new proposal in **Open UI**?

JS → Standard ES

CSS → CSS Profile?

i18n Resources



Coming soon: MiniApp Components; IoT

W3C Community Group Draft Report

MiniApp Common UI Components

Draft Community Group Report

Latest published version:
<https://www.w3.org/TR/miniapp-components/>

Latest editor's draft:
<https://w3c.github.io/miniapp-components>

Editor:
Zitao Wang (Huawei)

Copyright © 2021 the Contributors to the MiniApp Common UI Components Specification, published by the [MiniApps Ecosystem Community Group](#) under the [W3C Community Contributor License Agreement \(CLA\)](#). A human-readable [summary](#) is available.

Abstract

MiniApp UI component is the key to creating the MiniApp page. Each component builds an independent interactive function unit by simply encapsulating data and methods. This allows developers to reuse components in MiniApp to realize functions such as view containers, texts, and forms.

This document specifies a set of standard common UI components for MiniApp, that allows the developers to develop programs by combing these components. The standard MiniApp common UI component can help to improve the cross-platform capability of MiniApp and allows codes to easily migrate to different native apps and operating systems, improving developer work efficiency.

Status of This Document

This specification was published by the [MiniApps Ecosystem Community Group](#). It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the [W3C Community Contributor License Agreement \(CLA\)](#) there is a limited opt-out and other conditions apply. Learn more about [W3C Community and Business Groups](#).

[GitHub Issues](#) are preferred for discussion of this specification.

1. Introduction

1.1 An Overview of MiniApp UI Components

MiniApp is a concept of light software application, that can be distributed through any digital means, and accessed through the Web. Normally, MiniApps are made of high-level building blocks called Components, which allow developers to quickly construct the UI for a MiniApp.

MiniApp common UI components are usually stored in the common directory of the MiniApp package. Each component contains the following files:

- `.html`
html files describe the page layout.
- `.css`

W3C Community Group Draft Report

MiniApp Common UI Components

Draft Community Group Report

Abstract

Status of This Document

1. Background

2. Use Cases

- 2.1 Switch panel
- 2.2 Gateway
- 2.3 Smart Speaker
- 2.4 Checkout Pad
- 2.5 Video conference terminal
- 2.6 Face recognition terminal for access control
- 2.7 Smart home appliances
- 2.8 In-vehicle infotainment systems
- 2.9 Public information displays
- 2.10 Industrial HMI and control systems
- 2.11 Microcontroller kits

3. MiniApp for IoT Architecture

4. MiniApp Packaging for IoT

- 4.1 MiniApp Packaging for IoT device with a screen
- 4.2 MiniApp Packaging for IoT device without a screen

5. MiniApp Lifecycle for IoT

- 5.1 MiniApp Global Application Lifecycle for IoT
- 5.2 MiniApp Page Lifecycle for IoT

6. MiniApp APIs for IoT

- 6.1 MQTT API
- 6.2 Bluetooth Mesh API
- 6.3 UART (universal asynchronous receiver-transmitter) API
- 6.4 SPI (serial peripheral interface) API
- 6.5 GPIO (general-purpose input/output) API
- 6.6 I2C (inter-integrated circuit) API
- 6.7 Device low power management API
- 6.8 ADC (analog-to-digital converter) API
- 6.9 DAC (digital-to-analog converter) API
- 6.10 Hardware timer API

7. Privacy and Security



Figure 7 Home appliance user interface

2.8 In-vehicle infotainment systems

In-vehicle infotainment systems provide audio, video, navigation, and driving aids, as well as gaming entertainment for vehicle passengers. These systems may be deployed in private (e.g., car, van, and coach) and public transport (e.g., boat, train, bus, and aircraft) as a service for passengers.

Figure 8 shows an aircraft with infotainment displays in the back of the seats, allowing passengers to access in-flight information and interact with entertainment services. These systems, sometimes operating offline due to the limitations of the environment, could implement these infotainment services through MiniApps for IoT.



Figure 8 In-flight entertainment displays

2.9 Public information displays

Users may find public displays in different contexts. Smart cities deploy informative kiosks with maps, transport information, and other public services available to citizens and tourists in the streets, as shown in Figure 9. Similar informative screens may be found indoors in public facilities like airports, train stations, and shopping malls, sometimes allowing customers or users to interact with the system.

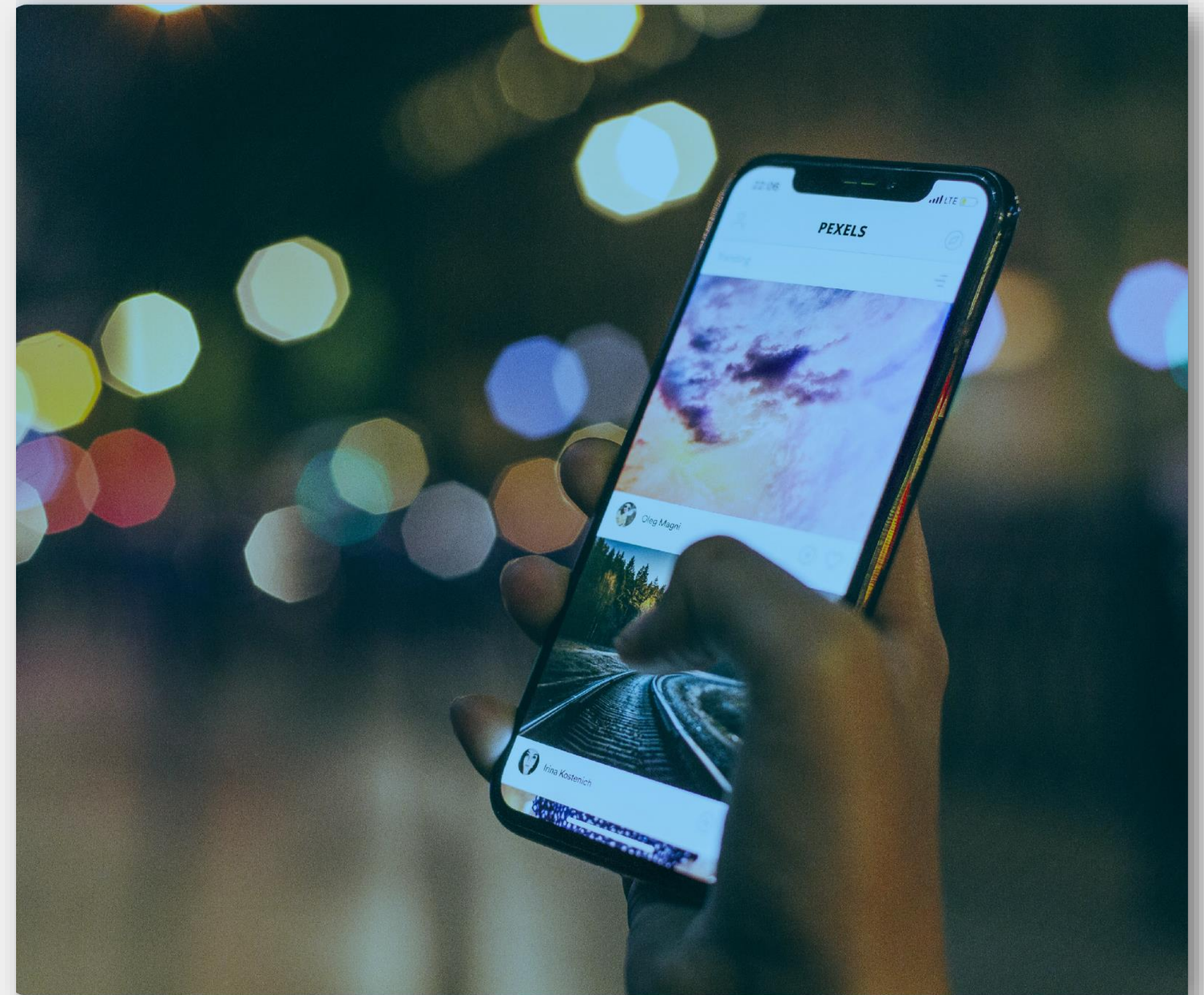
These public displays may include sensors and serve interactive services, applications, and widgets implemented through MiniApps for IoT. The MiniApp could access the sensors' information and offer user interaction to use its services.



Community

Raising awareness around Quick Apps

- Communication channels
- LinkedIn proposal
- Fireside chats
- Promo

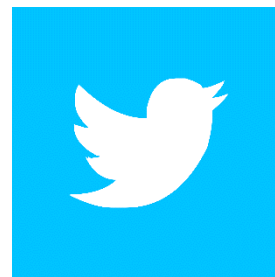


Communication channels brief strategy

Type of content:

- Questions, polls, discussions
 - News of the Initiative
- Events announcements
 - Infographics

TWITTER



Type of content:

- Post event articles
- Event announcements
- Editorials on industry topics
 - News and updates
 - Newsletters.

PORTAL



Type of content:

- educational (infographics, post about what quick apps are)
- reputational (new participants, success stories, use cases, etc.)
- informative (news, updates)
- interactive (polls, surveys)

LINKEDIN



Communication channels brief strategy

Type of content:

- recordings of fireside chats;
 - videos from events;
- short educational videos of what quick apps are.

YOUTUBE



- A few participants with a host who leads a discussion
- A mix of interview and an open discussion
- Topics are decided based on participants

TWITCH



2 of the most important events of the year are
OW2con
&
OW2 15 year anniversary

EVENTS



LinkedIn Page will help to:



- Engage with developers;
- Initiate new discussions in the comments and via polls, surveys, etc.
- Gather more data from developers, businesses and consumers;
- Generate more traffic to QAI website;
- Raise awareness about QAI.

Twitch & Fireside chat



Format:

- A few participants with a host who leads a discussion;
- A mix of interview and an open discussion;
- Topics are decided based on participants;

Frequency:

Once or twice per month

Topics:

Trending industry topics

Community actions starting soon

Fireside Chats

If you would like to become a guest or/and a host of future fireside chats, please reach out to:

yashchuk.valentyna@huawei.com
zachary.powell@huawei.com

LinkedIn

The poll is active till 25/02/2022
Please vote on LinkedIn proposal here:

<https://framadate.org/linkedinproposal>

Tour?

- Focus on W3C MiniApp standards
- Organised with W3C
- Spain, France, London, Germany?

QUICKAPP

What are Quick Apps?

Quick Apps are light native-like applications, or mini-apps, that do not require installation and run directly on the operating system, offering high performance and a rich user experience.

Native look and feel
Quick Apps offer native-like look and feel, as well as high running performance.

Easy to discover
Quick Apps can be found in app marketplaces, websites, banners and ads, widgets, referred by messages in social media or emails, and QR codes.

Installation free
Discover, discover a Quick App, you only need to tap on the link or scan the QR code to access and run the application. What a saver of the user time and device storage!

Global recognition
Quick Apps are supported by over 12 major smartphone manufacturers, and currently run on over 1.2 billion devices worldwide.

Easy to discover
Quick Apps may be listed in app marketplaces, websites, banners and ads, widgets, referred by messages in social media or emails, and QR codes.

Market recognition
Quick Apps are supported by over 12 major smartphone manufacturers, and currently run on over 1.2 billion devices worldwide.

QUICK APP

Quick Apps are based on W3C MiniApps standards, fostered by the MiniApp Working Group within the World Wide Web Consortium (W3C).

Infographic

Promo

QUICKAPP

What are Quick Apps?

Quick Apps are light native-like applications, or mini-apps, that do not require installation and run directly on the operating system, offering high performance and a rich user experience.

Native look and feel
Quick Apps offer native-like look and feel, as well as high running performance.

Easy to discover
Quick Apps may be listed in app marketplaces, websites, banners and ads, widgets, referred by messages in social media or emails, and QR codes.

Installation free
Once you discover a Quick App, you only need to tap on the link or scan the QR code to access and run the application. What a saver of the user time and device storage!

Global recognition
Quick Apps are supported by over 12 major smartphone manufacturers, and currently run on over 1.2 billion devices worldwide.

QUICK APP

Quick Apps are based on W3C MiniApps standards, fostered by the MiniApp Working Group within the World Wide Web Consortium (W3C).

Where Quick Apps can be used?

As light versions of potentially more sophisticated native apps, Quick Apps are excellent entry points to introduce customers to "full-fat" premium products.

Games

- Easy to discover and play
- May include Ads and IADs
- One-click authentication
- Trial, preview version

Bars & Restaurants

- Touchless service
- QR code and access menu
- Targeted discounts
- Book, order, pay

Health

- Emergencies, first aid
- Nearby facilities & services
- Alert notifications
- Exposure warnings

Travel, information

- Booking on the go
- QR-code, book and pay
- In-journey information
- Content based on location
- Real-time alerts

Utilities

- Touchless, online services
- Covering citizens' needs
- Utility, taxes payments
- Smart city productivity

Brochure

QUICKAPP

What are Quick Apps?

Quick Apps are light native-like applications, or mini-apps, that do not require installation and run directly on the operating system, offering high performance and a rich user experience.

Native look and feel
Quick Apps offer native-like look and feel, as well as high running performance.

Easy to discover
Quick Apps may be listed in app marketplaces, websites, banners and ads, widgets, referred by messages in social media or emails, and QR codes.

Installation free
Once you discover a Quick App, you only need to tap on the link or scan the QR code to access and run the application. What a saver of the user time and device storage!

Global recognition
Quick Apps are supported by over 12 major smartphone manufacturers, and currently run on over 1.2 billion devices worldwide.

QUICK APP

Quick Apps are based on W3C MiniApps standards, fostered by the MiniApp Working Group within the World Wide Web Consortium (W3C).

Where Quick Apps can be used?

As light versions of potentially more sophisticated native apps, Quick Apps are excellent entry points to introduce customers to "full-fat" premium products.

Games

- Easy to discover and play
- May include Ads and IADs
- One-click authentication
- Trial, preview version

Bars & Restaurants

- Touchless service
- QR code and access menu
- Targeted discounts
- Book, order, pay

Health

- Emergencies, first aid
- Nearby facilities & services
- Alert notifications
- Exposure warnings

Travel, information

- Booking on the go
- QR-code, book and pay
- In-journey information
- Content based on location
- Real-time alerts

Utilities

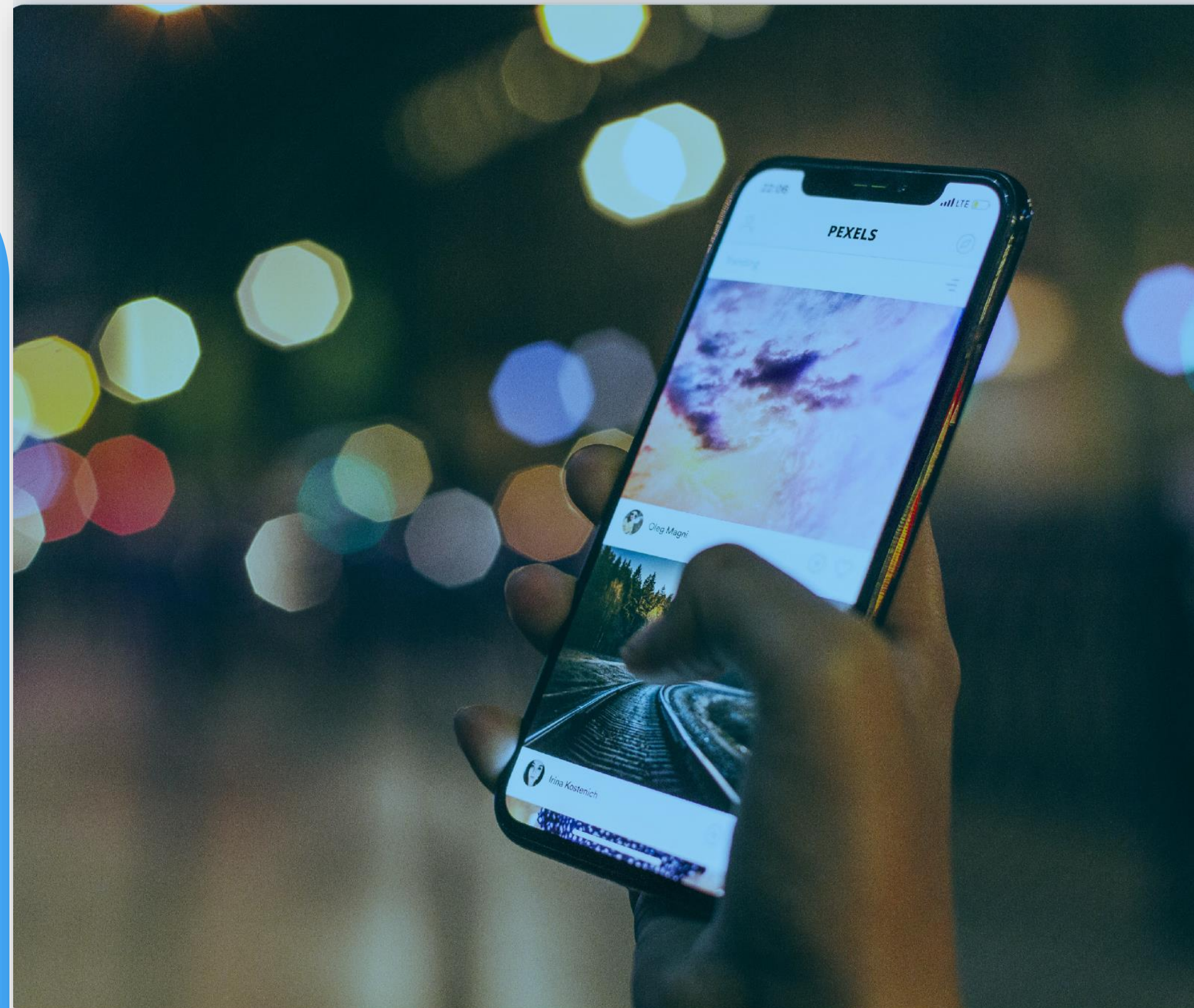
- Touchless, online services
- Covering citizens' needs
- Utility, taxes payments
- Smart city productivity

Slides



Workgroups

Task Forces, studies, POCs, pilots & collab projects, Discussion groups



WORKGROUPS

GAMING

- FRVR + Famobi

SUSTAINABILITY

- Alliance Tech
- Fashion supply chain project proposal
- ?? Change name to "Industry 4.0" ??



EDUCATION

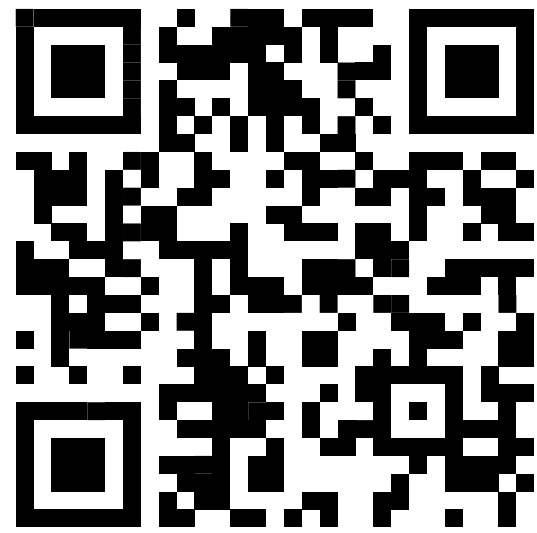
- Needs a new coordinator
- Contact with **Telanto** being established
- Quick Apps for inclusion?

RUSSIA

- Tinkoff
- Yandex
- SBER

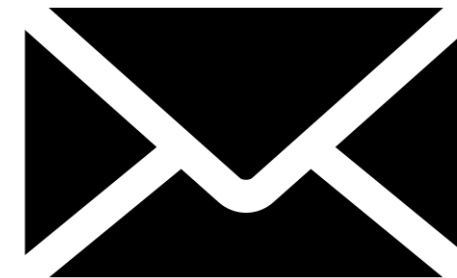


OW2 Quick App Initiative



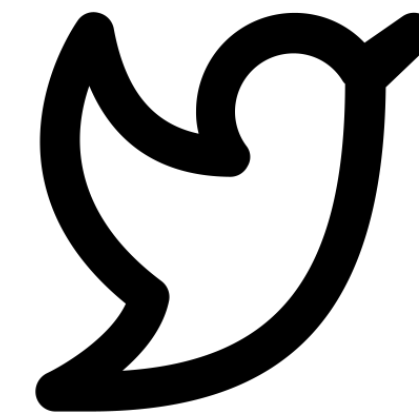
DISCOVER

<https://quick-app-initiative.ow2.io>



CONTACT

quickapp-team@ow2.org



FOLLOW

[@OW2QuickApps](#)
[#QuickAppsEU](#)



Acknowledgements

Quick App Initiative Participants
