

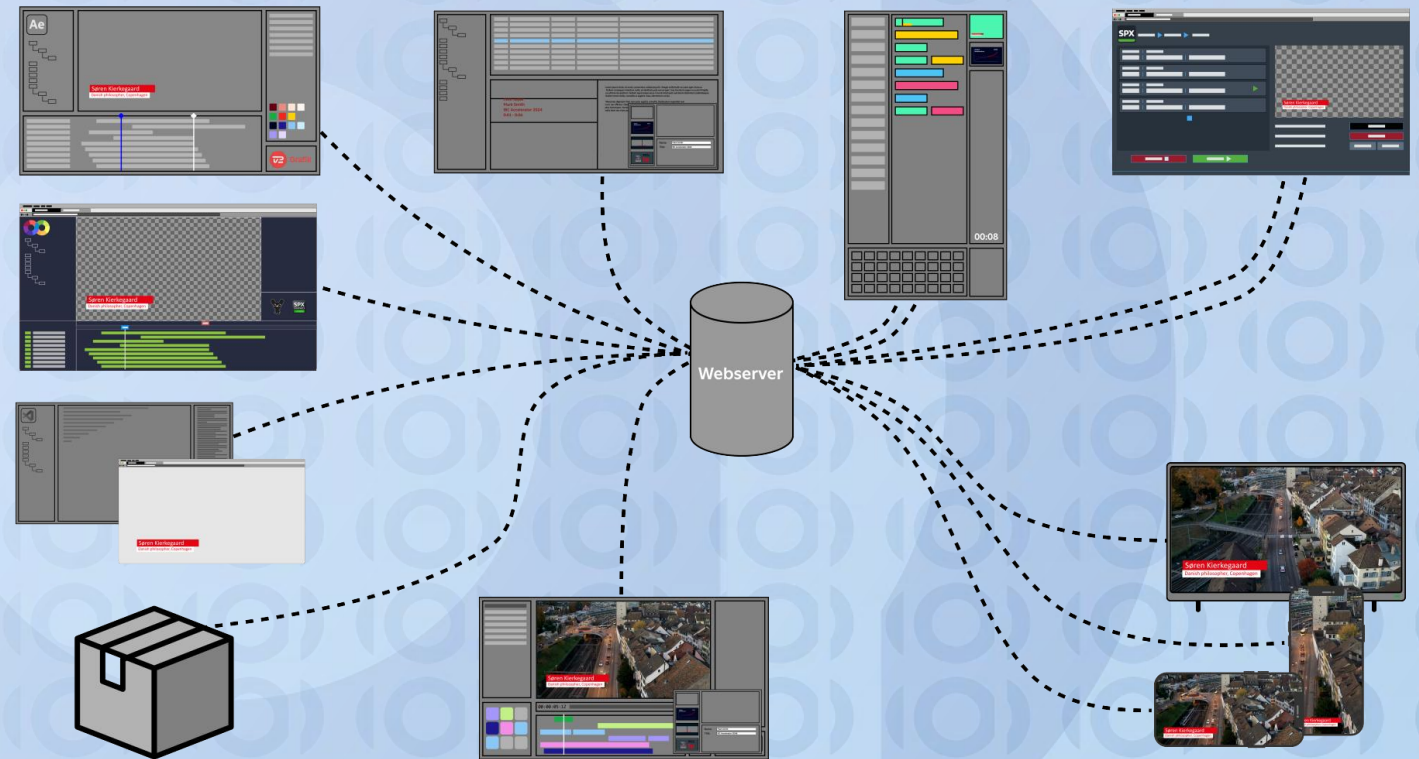
ograf

- Open HTML Graphics standard



AGENDA

- Problem & Challenge
- Target & Solutions
 - Architecture
- Demo
- Technical details
 - Samples
- Stakeholders
- Roadmap



HTML GRAPHICS
FINDING A COMMON STANDARD

Problem & Challenge

PROBLEM

- All broadcast graphics providers deliver complete end-to-end solutions, lacking the ability to integrate different systems.
- Every provider employs unique binary file formats.
- There is no capability for rendering on end-user devices.



THE CHALLENGE

How standards proliferate:
(See: A/C chargers, video formats, instant messaging, etc.)

Situation:

There are 14
Competing standards

14?! RIDICULOUS!

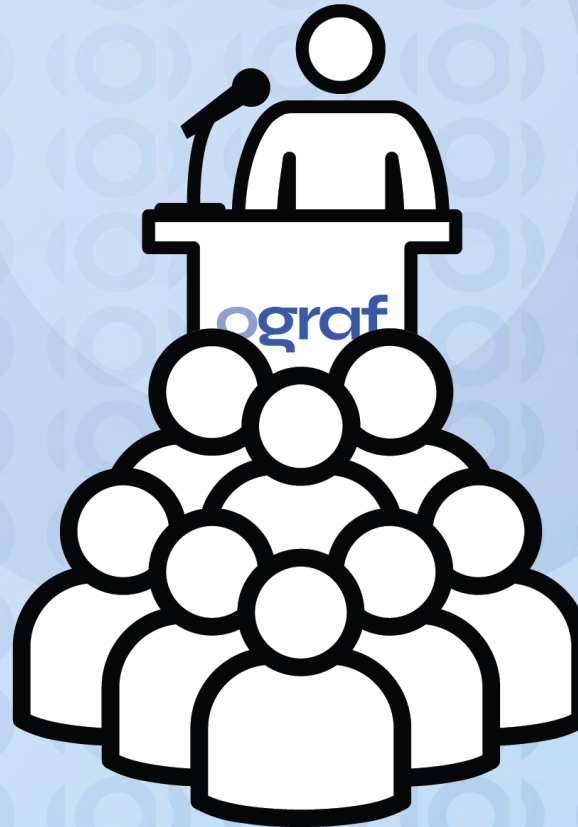
We need to develop a universal
standard that covers everyone's
use cases.

New

Situation:

There are now 15
competing standards!

But we should avoid the
need for a 16th to appear!



HTML GRAPHICS
FINDING A COMMON STANDARD

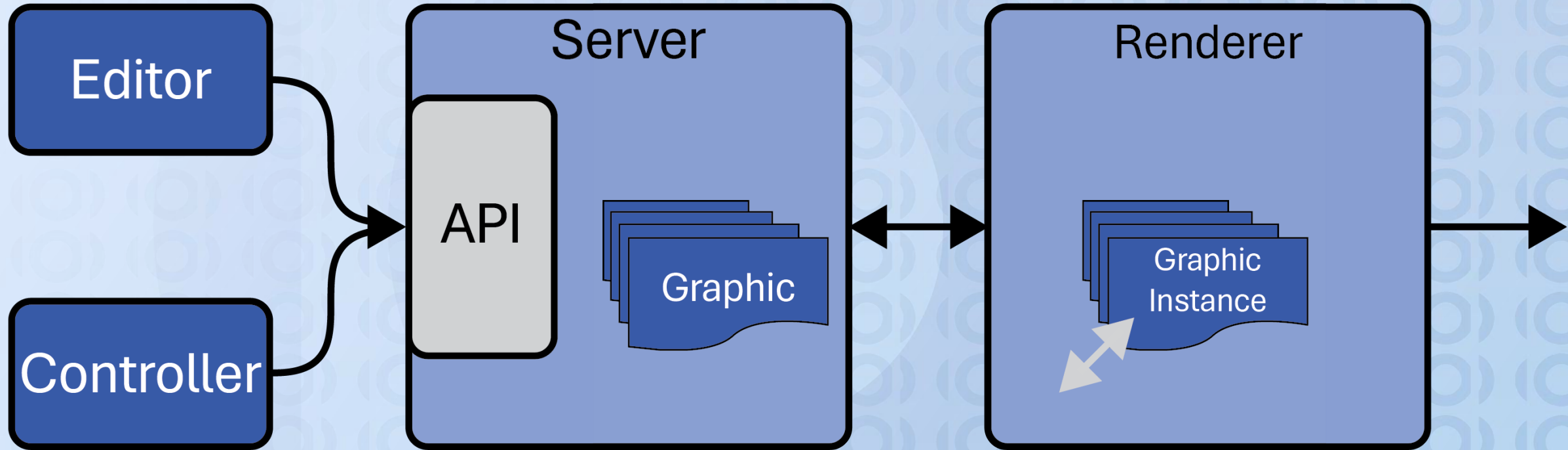
Target & Solution

SOLUTION

- A single open HTML-based standard
- Descriptions of workflows
- Detailed documentation
- Clear instructions to follow
- Sample graphics packages
- Endorsement and support from numerous vendors and broadcasters



WEB-BASED GRAPHICS WORKFLOW

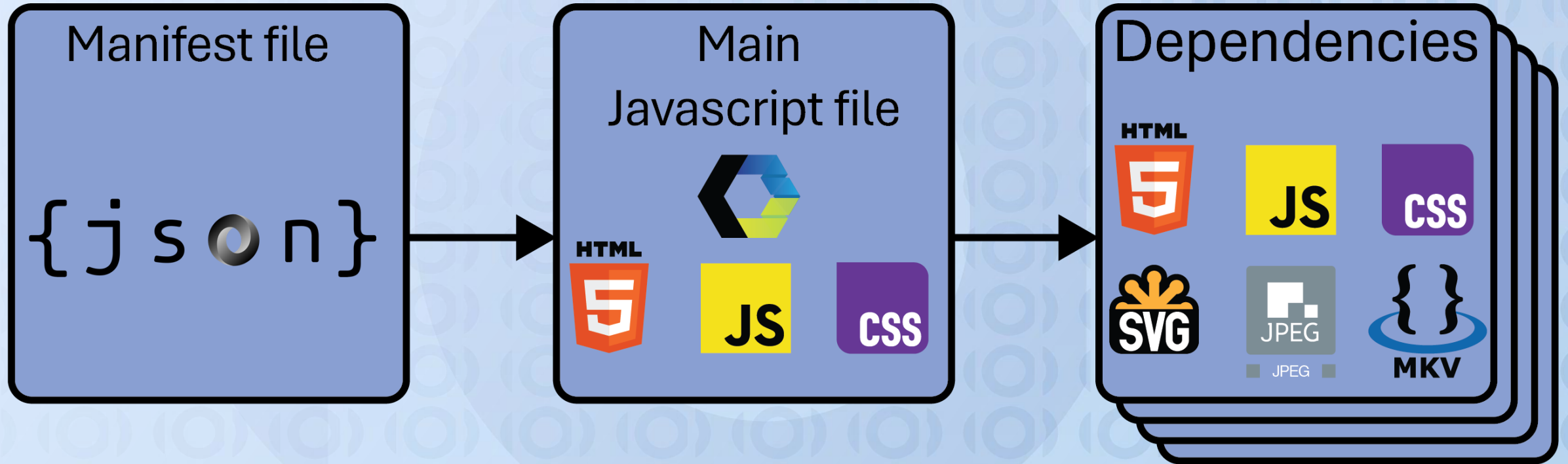


Scope of the OGrاف Specification

SPECIFICATION REQUIREMENTS

- Create a Web-based Graphic once, use it in many Renderers
- Unopinionated in terms of Web technologies
- Backwards and forwards compatible
- Extensible with vendor-specific functionality
- Support for realtime and non-realtime Graphics

ANATOMY OF AN OGRAF



JSON file
describing the Graphic

Javascript file
with Graphic Web Component

HTML GRAPHICS
FINDING A COMMON STANDARD

Graphics manifest

MANIFEST FILE

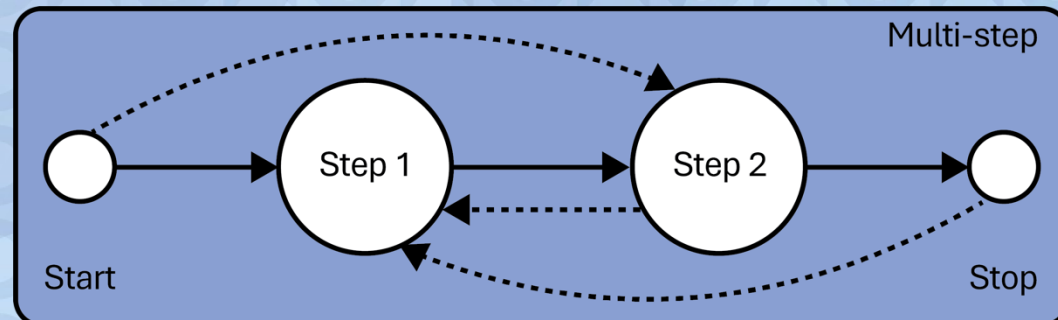
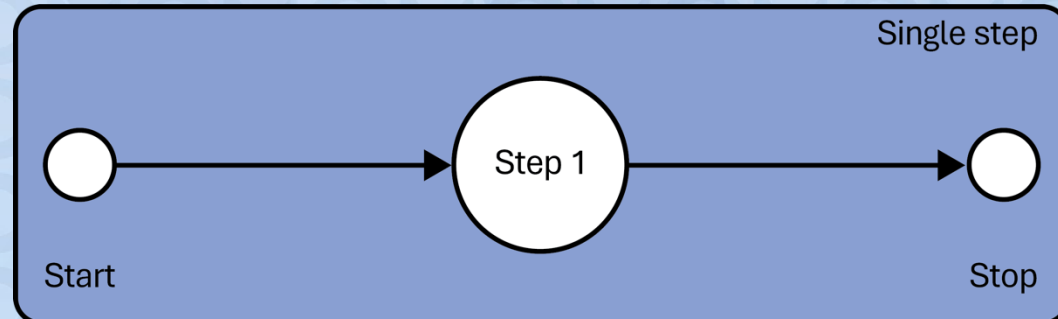
- JSON file containing metadata about the Graphic
- MUST be compliant with the JSON Schema of the Manifest model
- No restrictions on the file name
- Can be seen as the entrypoint and representation for the Graphic

MANIFEST FILE

Category	Fields
Unique identifier	id*
Metadata	name*, description, author, version
Main Javascript file	main*
Rendering properties	supportsRealTime*, supportsNonRealTime*, stepCount
Rendering requirements	renderingRequirements
Schema for internal state	schema
Custom actions	customActions

STEP MODEL

- Step = state where Graphic is ‘paused’
 - paused $\neq$ nothing is moving
- Transitions between steps
 - externally triggered
 - animated or not



CUSTOM ACTIONS

- action that is only known within the current Graphic
- described in the manifest to support auto-generated user/machine interfaces
- consists of
 - **id***
 - **name***
 - **description**
 - **schema**
 - JSON Schema for the action parameters

VENDOR-SPECIFIC FIELDS

- Additional fields that are not part of the OGraf specification
- Can occur in the Manifest file
- Can occur in the parameters and responses of the actions
- MUST be prefixed with **v_**
 - for example: **v_editor**

HTML GRAPHICS
FINDING A COMMON STANDARD

Graphics implementation

MAIN JAVASCRIPT FILE

- Referenced from the Manifest
- Entrypoint file for the Graphic implementation
 - will be loaded by the Renderer
- Exposes a standardized interface towards Renderers

GRAPHIC INTERFACE

load()	called when Graphic is loaded inside the Renderer
dispose()	called before Graphic is removed from the Renderer
playAction()	play a given step, with or without animation
stopAction()	stop a Graphic, i.e. bring it to the end state, with or without animation
updateAction()	update one or more fields of the internal state of the Graphic
customAction()	invoke a custom action on the graphic
gotoTime()	jump to a certain timestamp, expressed in milliseconds
setActionSchedule()	schedule actions to be invoked at a certain point in time



WEB COMPONENTS

- A suite of different **technologies** allowing you to create **reusable custom elements**, with their functionality fully encapsulated
 - Custom Elements
 - Shadow DOM
 - HTML Template
- Standardized technologies, native support from browsers
- Supported by existing frameworks like Vue, React, etc.

<https://www.webcomponents.org/>

<https://html.spec.whatwg.org/multipage/custom-elements.html#custom-elements>

GRAPHIC INTERFACE AS A WEB COMPONENT

- A Graphic MUST be implemented as a Custom Element
 - Javascript class that extends **HTMLElement**
- Class MUST implement the functions
 - load, dispose, playAction, stopAction, updateAction, customAction
 - gotoTime, setActionSchedule (for non-real-time only)
- Default export MUST be used to export the class

REQUIREMENTS FOR A RENDERER

- To add a Graphic
 - import the Graphic into the Renderer
 - MUST call the **load()** function
 - MUST wait for this function to complete before any other actions can be executed on the Graphic
- To remove a Graphic
 - MUST call the **dispose()** function
 - SHOULD wait for this function to complete before removing the Graphic from the Renderer

HTML GRAPHICS
FINDING A COMMON STANDARD

Demo

The background is a solid blue color with a repeating pattern of small, light blue circles. A large, stylized, light blue 'C' shape is centered on the page, partially obscuring the text.

Into the details

DOCUMENTATION & TOOLS

All documentation and links to tools can be found at

<https://ograp.ebu.io>

HTML GRAPHICS
FINDING A COMMON STANDARD

Stakeholders

STAKEHOLDERS



ograft Roadmap

March 2025:

- First public draft of Graphics API

September 2025:

- Version 1 published of Graphic API
- First public draft of Rendering System API

End of 2025:

- Version 1 extended with Rendering System API

THE CHALLENGE

How standards proliferate:
(See: A/C chargers, video formats, instant messaging, etc.)

Situation:

There are 14
Competing standards

14?! RIDICULOUS!

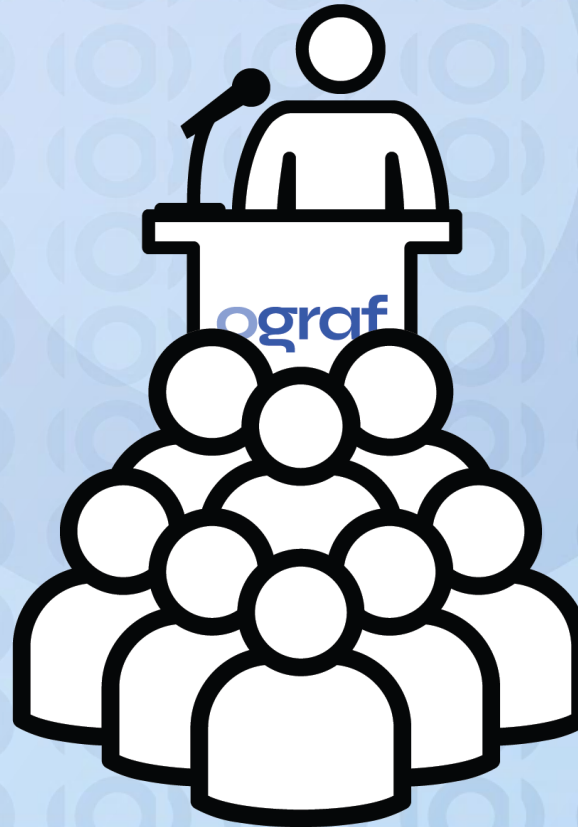
We need to develop a universal
standard that covers everyone's
use cases.

New

Situation:

There are now 15
competing standards!

But we should avoid the
need for a 16th to appear!



Questions & Comments?

Thank you

ograf.ebu.io

TUOMO KULOMAA
SPX
TUOMO@SOFTPIX.IO

IVAN MARETIC
LOOPIC
INFO@LOOPIC.IO

DAVY VAN DEURSEN
NETON.LIVE
DVD@NETON.LIVE

JOHAN NYMAN
SUPERFLY.TV
JOHAN@SUPERFLY.TV

NIELS BORG
TV 2 DANMARK
NIBB@TV2.DK
+45 2298 4181