ArcGIS API for Javascript Road-ahead

Yann Cabon - Jeremy Bartley

3.x

- Geometry Engine
- Smart Mapping
- Image Server
- Quantization vs. Generalization

4.0 - highlights

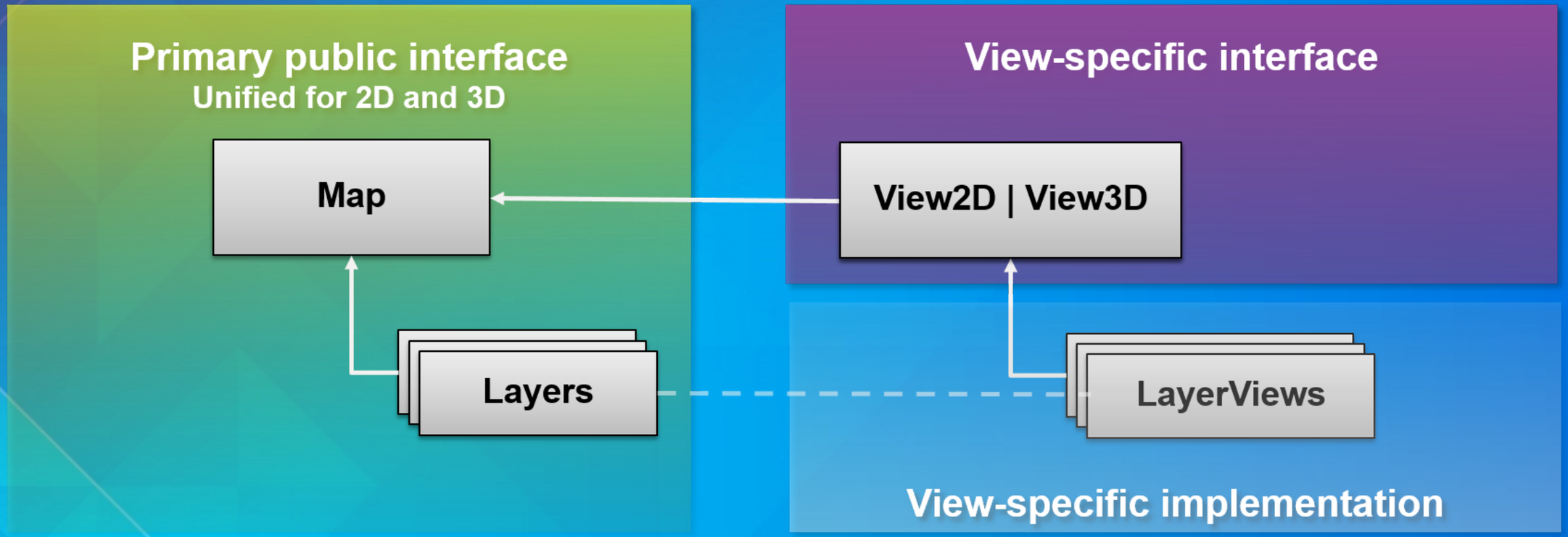
- 2D/3D
- 4.0beta1 released July 15th!
- multiple betas coming
- API 4.0: new concepts & changes
- IE9+ for 2D, IE11+ for 3D

2D/3D

- Starting point of 4.0: 3D is coming!
- currently in 3.x:
  - Map, many DOM nodes
  - Each Layer, 1 DOM Node
- Can't work, WebGL renders in one Canvas
- Solution?

2D/3D

- Separate the business logic from the drawing logic.

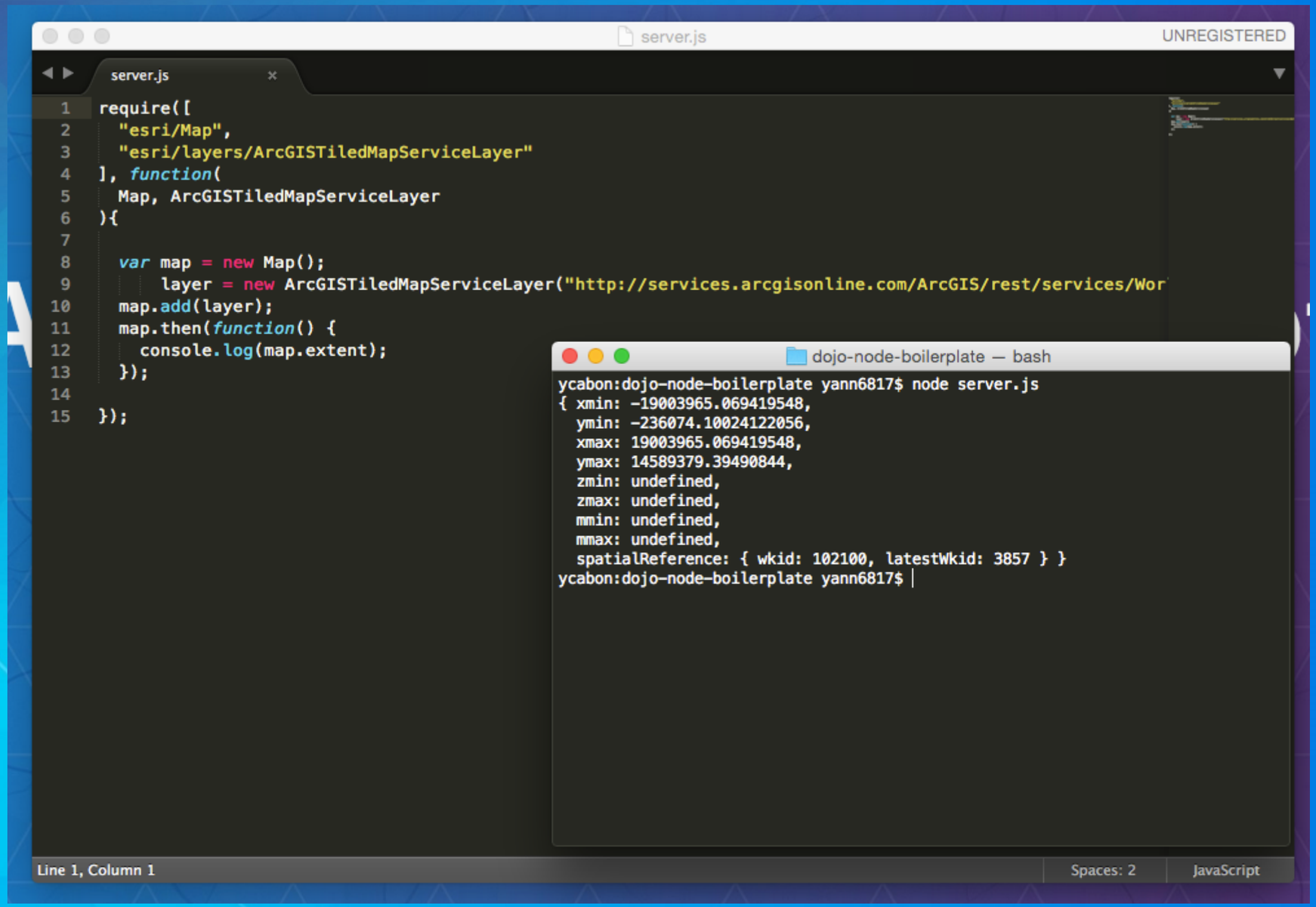


- Communication model by **events** and **properties watching**
  - clean decoupling
  - clearer about what's going on when something changes

2D/3D

- For the rest, one API
- demo

Experiment - Map running in node



```
// create the map and its layers
var map = new Map({
  basemap: "topo"
});
map.add(new FeatureLayer(...));

// create a 3D view for the Map
var view = new SceneView({
  map: map,
  container: "viewDiv"
});
```

esri/Accessor

- Mixin similar to `dojo/Stateful`
- single object constructor
- `get()`, `set()`, `watch()`

```
map.watch('basemap', function(newValue, oldValue, name, target) {
  // ...
});
```

- support for ES7 `Object.observe()`

Properties watching

- Direct benefits:
  - remove inconsistancies between constructor, getter, setter functions, events
  - one convention everywhere. *"just need to know what properties for a class"*
  - Single object constructor, no more 3+ constructors
  - Leaner SDK: we doc only the properties, the rest is convention
- Changes:
  - no more ***property***-change events, use `watch()`
  - in 3.x, listen for `extent-change` event.
  - in 4.0 `extent` watchers will be call very often
  - new events and properties for animation.

Properties watching

- Frameworks integration
  - properties are framework agnostic
  - better/easier integration
- Examples
  - side by side views
  - dbind
  - React
  - camera recorder

Layers

- `map.layers`, a collection of the operational layers
  - mix of image AND graphics
- Shorter names: `ArcGISTiledLayer`, `ArcGISDynamicLayer`
- new ones:
  - `ArcGISElevationLayer`
  - `SceneLayer`
  - `GroupLayer`

GroupLayer

- New layer: GroupLayer
- group layers together
- structure your data visualization
- visibility mode: `exclusive`, `independent`, `inherit`
- listMode: `hide-children`, `hidden`
- demo

Collection

- More or less like an Array
- `add` / `remove` / `forEach` / `map` / `find` / `findIndex`...
- emit "change" events when something is added/removed/moved
- used for layers, used for layers in Basemap, used for graphics...

Basemap

- full fledge class `esri/Basemap`
- basemap's layers are not part of the `map.layers`, but from `map.basemap`
- contains 3 Collections: baseLayers, referenceLayers, elevationLayers
- can be set with
  - string for esri's basemap
  - or custom Basemap instance
  - in 2D and 3D

Basemap

- `basemap` as a string, creation of the appropriated Basemap instance

```
var map = new Map({
 basemap: 'topo'
});

map.basemap = 'streets';
```

- `basemap` as an instance of `Basemap`

```
var map = new Map({/*...*/});

var toner = new Basemap({
 baseLayers: [
   new WebTiledLayer({
     urlTemplate: '...'
   })
 ]
})

map.basemap = toner;
```

2D

- new "engine" in the work.
- faster, more future proof
  - abstraction to draw tiles and dynamic images to ease custom layers/layerviews
  - abstraction to draw in DOM or Canvas, possibly webgl ;-)
- display graphics while zooming.
- rotation
- continous zoom
- vector map tiles, basemaps

3D

- webgl engine to display the earth.
- z/m support in the API, tasks, layers...
- support for simple symbols
- new 3D Symbols

Resizing logic

- automatically measure and position the view
- resize by center, or not
- better integration with responsive design pages
- and frameworks

Padding

- easier fullscreen view application.
- defines inner margin to make space for UI.
- 2D
- 3D

Animation

- generic function `animateTo(target, options):Promise`
- customize easing, duration, chaining
- DIY using other libs or custom
- `esri/Viewpoint`: common way to share between 2D/3D

Widgets

- `ui` property on view to quickly place components
- widgets designed as MVVM
  - separates the logic from the UI implementation
  - easier to create new versions using other frameworks
- ported to 4.0beta1: Search, Zoom, Attribution
- new ones: Compass

WebMap & WebScene APIs

- read
- save / save as
- easier portal / arcgis.com interaction

SDK

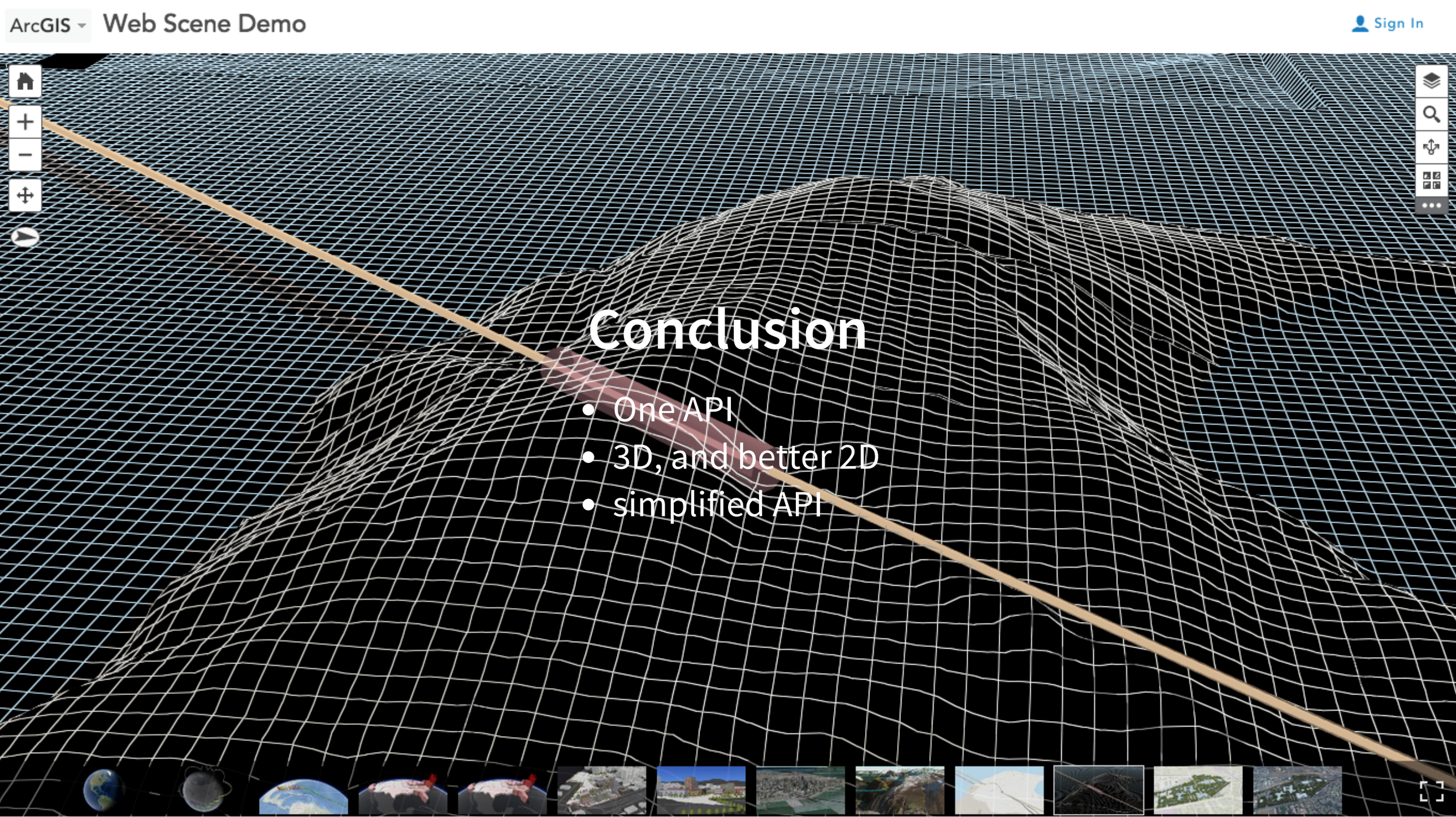
- new SDK, built from scratch
- simpler, focused samples
- user experience
- more code snippets

Other

- legacy dojo loader removed - AMD only
- classes properly cased: esri/Map, esri/Graphic, esri/layers/Layer
- new folder structure.

Beta2

- initial implementation of WebMap and WebScene. reading first
- more layer support
- more widgets
- performance improvements
- 3D
  - realistic atmosphere
  - subsurface rendering
  - point cloud
  - planar mode
- bugs...


ArcGIS
Web Scene Demo
Sign In
Conclusion
• One API
• 3D, and better 2D
• simplified API

Questions