



# Cross-platform native components using fundamental-styles

and couple of other tools\*

---

Giorgi Cheishvili, Software developer  
Valor Software



How do we support multiple  
environments **efficiently**?

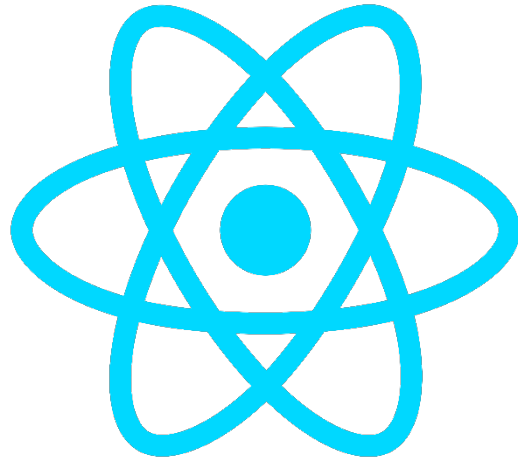


fundamental conf •



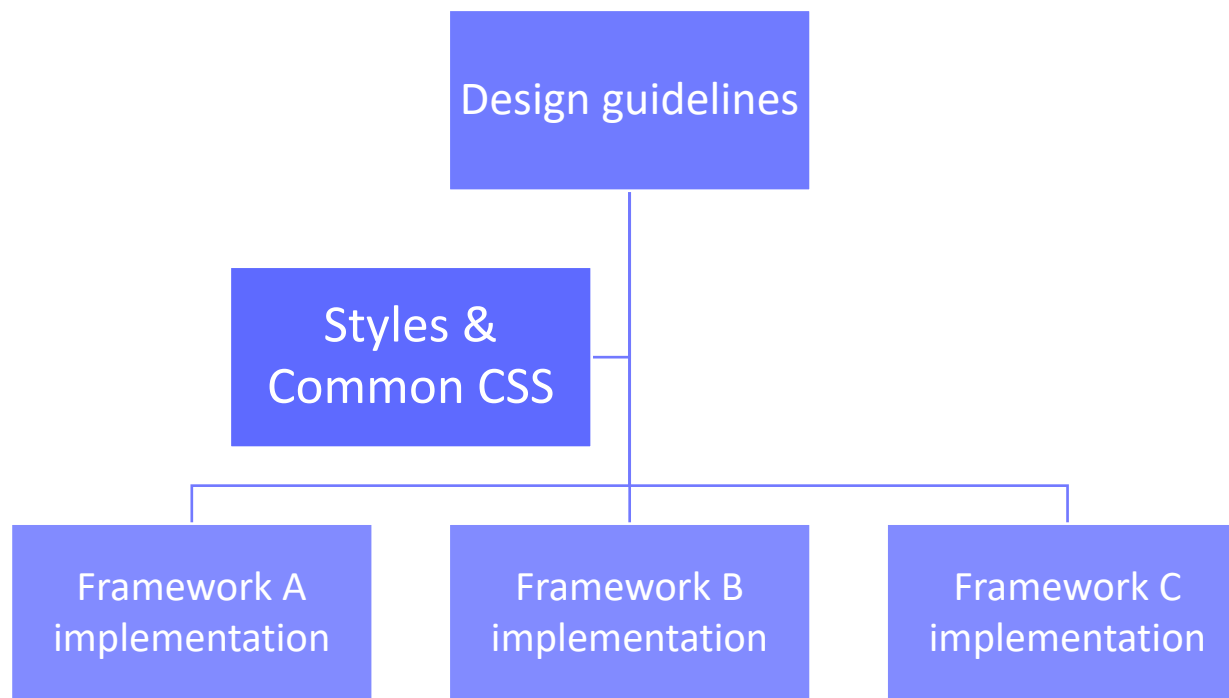
- Extensive tooling
- Great Forms
- SSR out of the box
- Dependency injection
- Progressive Web Apps
- Clear style guides and practices
- Huge community
- Flexible
- Many more things...



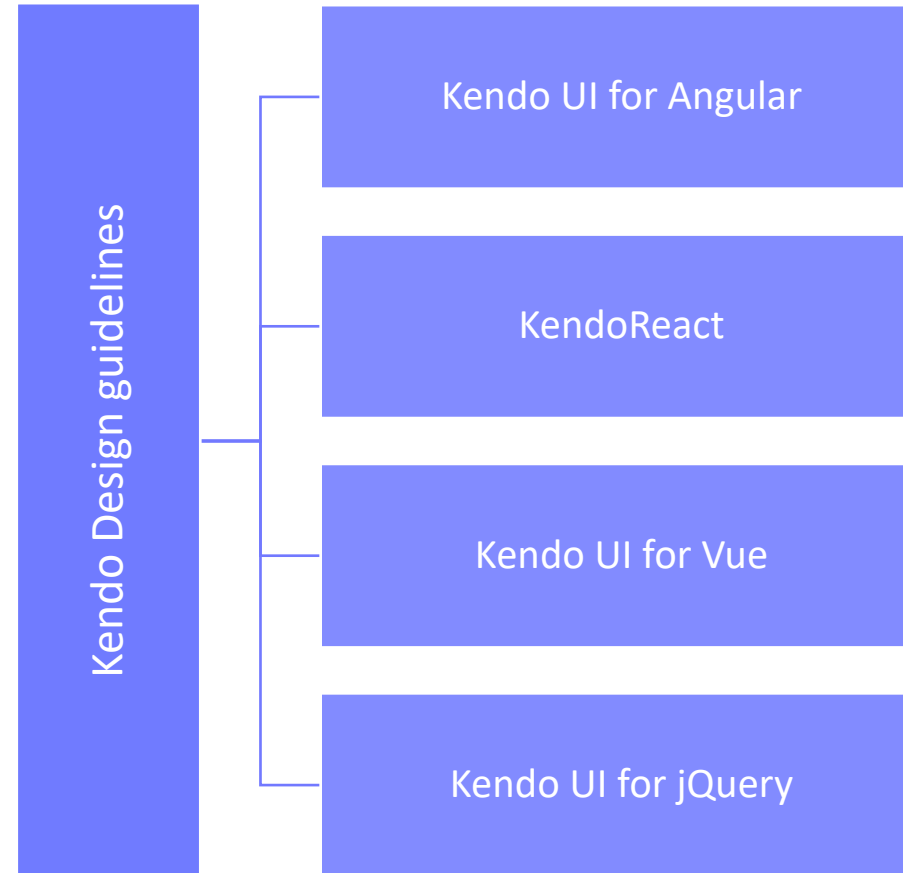


- Easy to use
- High performance
- Very flexible
- Extensive amount of 3<sup>rd</sup> party libraries
- Huge community
- Diversity of paradigms
- Probably more things...

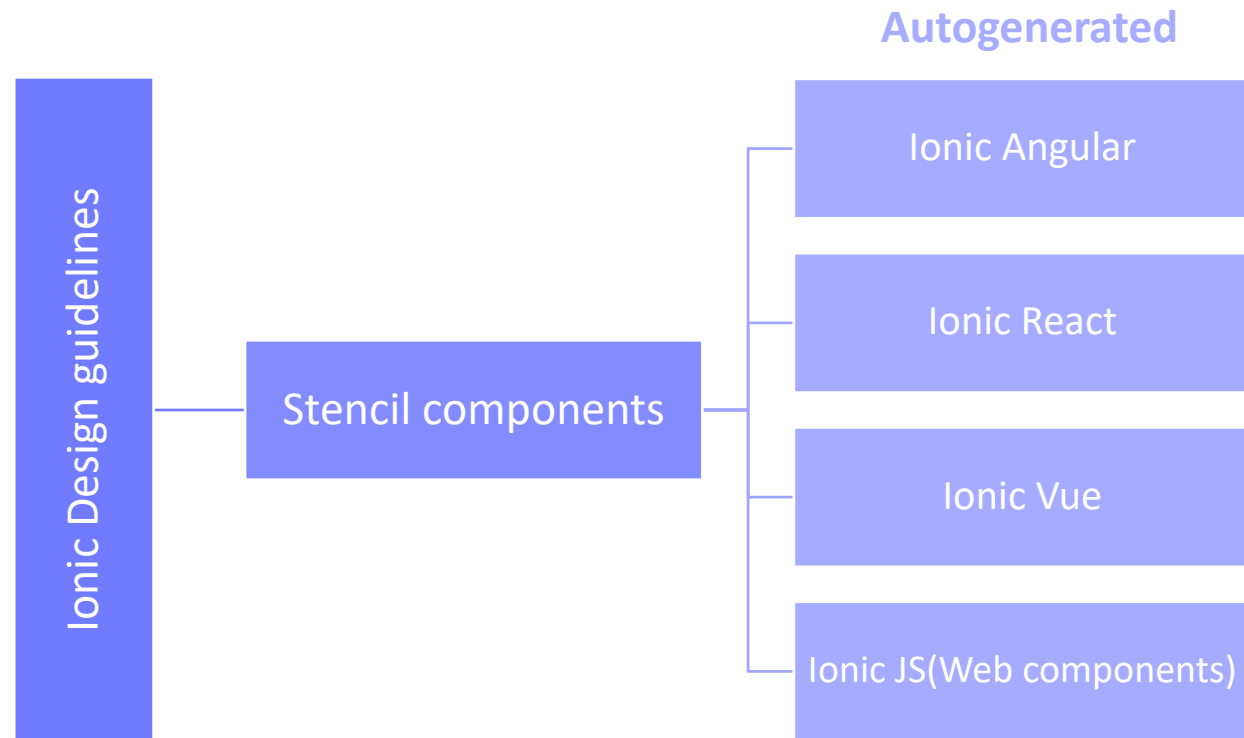




# Kendo UI products' structure



# Ionic products' structure



Solution(?)

Have universal compiler, of course



Fundamental conf





# Mitosis

Write components once, compile to every framework:



fundamental conf

# Mitosis.JS JSON representation of file contents



```
1 {
2   "@type": "@builder.io/mitosis/component",
3   "imports": [],
4   "exports": {},
5   "inputs": [],
6   "meta": {
7     "useMetadata": {
8       "somethingSpecialAboutThisComponent": true
9     }
10  },
11  "refs": {},
12  "state": {},
13  "children": [
14    {
15      "@type": "@builder.io/mitosis/node",
16      "name": "h1",
17      "meta": {},
18      "scope": {},
19      "properties": {},
20      "bindings": {},
21      "children": [
22        {
23          "@type": "@builder.io/mitosis/node",
24          "name": "div",
25          "meta": {},
26          "scope": {},
27          "properties": {
28            "_text": "Hello "
29          },
30          "bindings": {},
31          "children": []
32        },
33        {
34          "@type": "@builder.io/mitosis/node",
35          "name": "div",
36          "meta": {},
37          "scope": {},
38          "properties": {},
39          "bindings": {
40            "_text": {
41              "code": "props.name || 'World'"
42            }
43          },
44          "children": []
45        },
46        {
47          "@type": "@builder.io/mitosis/node",
48          "name": "div",
49          "meta": {},
50          "scope": {},
51          "properties": {
52            "_text": "!"
53          },
54          "bindings": {},
55          "children": []
56        }
57      ]
58    },
59  ],
60  "hooks": {},
61  "context": { ...
62  },
63  "name": "MyComponent",
64  "subComponents": [],
65  "types": [
66    "interface MyComponentProps {\n  name: string;\n}"
67  ],
68  "propsTypeRef": "MyComponentProps"
69 }
70
71 }
```

nx: ^14.0.0  
@builder.io/mitosis: ^0.0.58  
fundamental-styles: ^0.24.4



## Problem with compiling to Angular target



There is no generated module...



## Problem with compiling to Angular target



There is no Angular form support...



## Problem with compiling to Angular target



There is no grouping of components into modules, because there is no modules...



But there are plugins!

```
const mitosisPlugin = (options: any) => {
  return {
    json: {
      pre: (json: MitosisComponent): MitosisComponent => {
        // You can do anything with it.
        // Add imports or remove them
        // Update name, selector, exports,
        // all sorts of things and it will
        // be called before generator starts it's magic
      },
      post: (json: MitosisComponent): MitosisComponent => {
        // same capabilities here, just after generator's
        // modifications
      }
    },
    code: {
      pre: (beforePrettierContent: string): string => {
        // Code can be modified here
      },
      post: (afterPrettierContent: string): string => {
        // Code can be modified here
      }
    }
  }
}
```



fundamental conf



Lets go to the code





Fictional(probably based on real events somewhere) story and imaginary world, where lion and the sheep eat grass together



## Dictionary

Definitions from [Oxford Languages](#) · [Learn more](#)



# epilogue

/ˈɛpɪlɒɡ/

*noun*

a section or speech at the end of a book or play that serves as a comment on or a conclusion to what has happened.

"the meaning of the book's title is revealed in the epilogue"

**Similar:**

afterword

postscript

PS

coda

codicil

appendix

tailpiece




- a final or concluding act or event.  
"a disastrous **epilogue to** the rest of his career"




# Thank you for your attention!

If you have any questions, ideas or something to share, you can always find me at:

 [giorgi-cheishvili](#)

 [giorgi.cheishvili@valor-software.com](mailto:giorgi.cheishvili@valor-software.com)

 [@g-cheishvili](#)

