The reality of C++ combined JavaScript for modern GUI development

Markus Klemm
www.markusklemm.net

For people skipping/searching

- CMake based, no gyp
- Integration of non JS aware native projects and binaries or drivers
- React app frontend without ejection
- Electronjs frame
- Tested in production for windows linux and mac
- https://github.com/Superlokkus/electron_gui
- https://github.com/nodejs/node-addon-api/issues/803
- https://github.com/nodejs/node-addon-api/pull/804

Up next

- Motivation
- Project proclamation
- NAPI
- Hands on C++
- Pitfalls
- C++ language retrospective

Trends and challenges in portable GUI development

- Different render or operating system interfaces
- Different resolutions
- Different DPI
- Different aspect ratios
- Different native inputs (touch, dials, native UI look)

New

27-inch model

iMac

The all-in-one for all.

Buy

Learn more ›

New

27-inch model

iMac

The all-in-one for all.

Buy

Learn more ›

Reaction in Qt

- Before Qt5, one wrote subclasses of widget classes which
  - Created sub widgets, usually as a side effect in the constructor
  - Changed their own properties and the properties of their children
- Done in C++, in a imperative way independent of visible UI structure

```
auto toolbar = new QToolBar;
this->addToolBar(toolbar);

auto person_action = toolbar->addAction(QString::fromStdWString(L"Personen"));
auto book_action = toolbar->addAction(QString::fromStdWString(L"Bücher"));
auto dvd_action = toolbar->addAction(QString::fromStdWString(L"DVDs"));
auto lent_mediums_action = toolbar->addAction(QString::fromStdWString(L"Ausgeliehene Medien"));

auto lend_action = toolbar->addAction(QString::fromStdWString(L"Leihe aus"));
auto give_back_action = toolbar->addAction(QString::fromStdWString(L"Gib zurück"));

connect(person_action, SIGNAL(triggered()), this, SLOT(show_persons()));
connect(book_action, SIGNAL(triggered()), this, SLOT(show_books()));
```

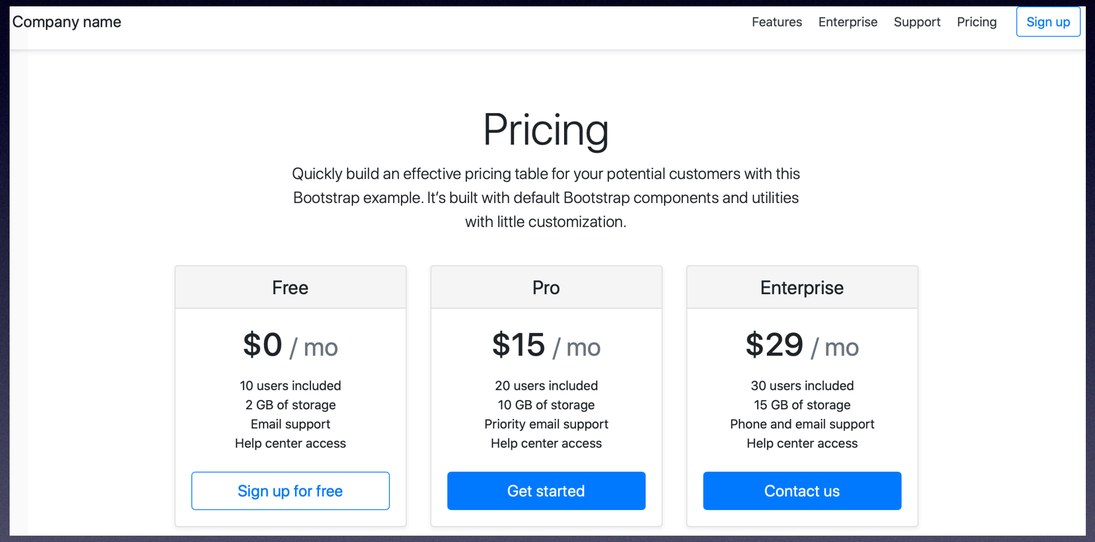
Reaction in Qt

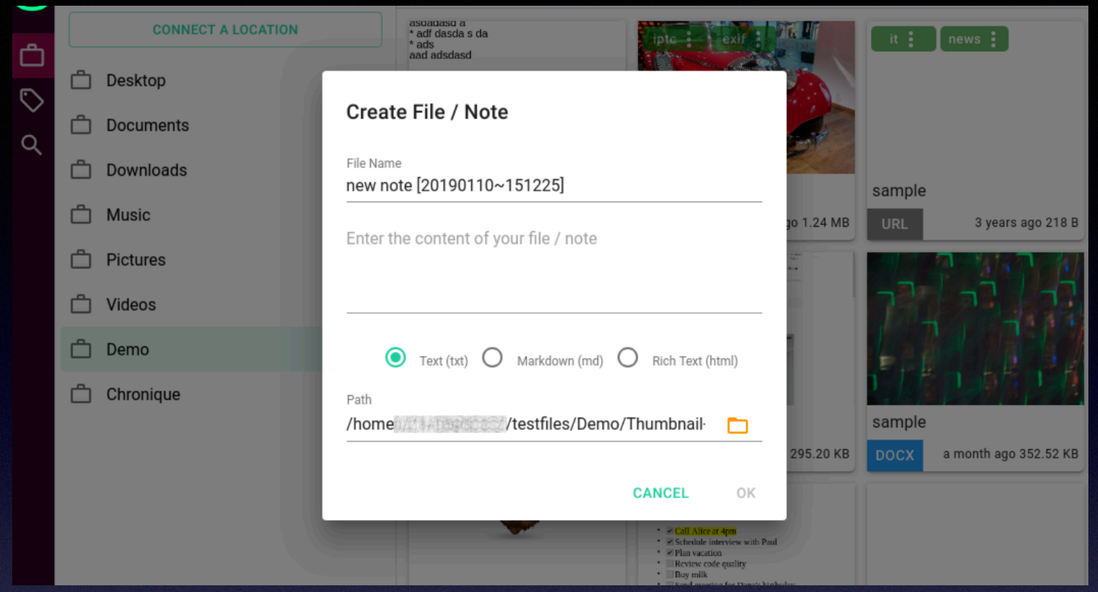
- Productive Introduction of QML
- Markup language
- JSON dialect
- Fast change, no recompile
- Reflects UI structure

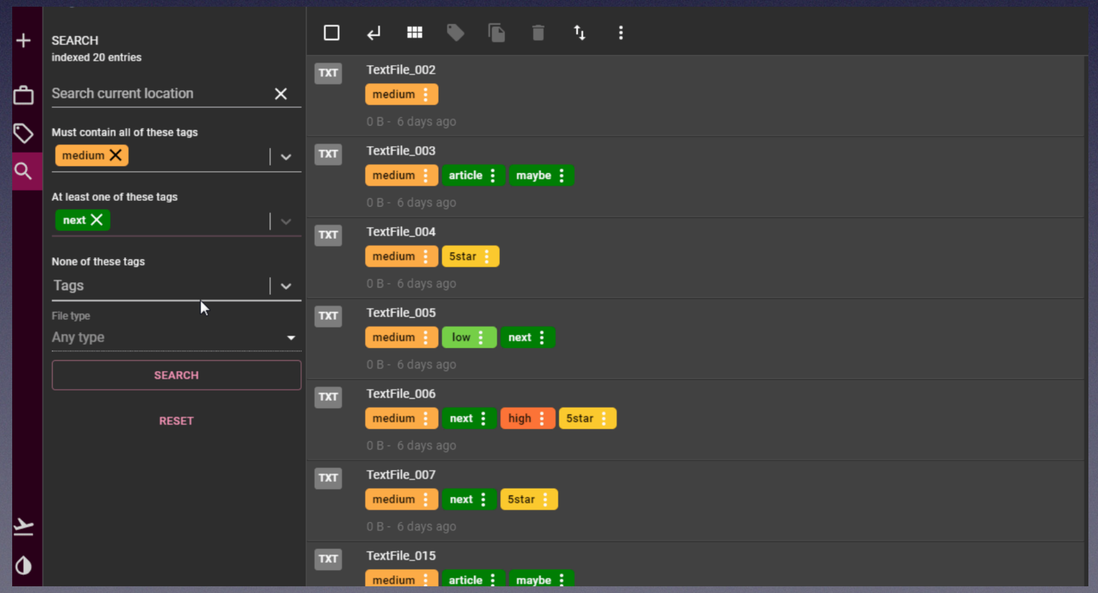
```
import QtQuick 1.0

 Rectangle {
     id: simplebutton
     color: "grey"
     width: 150; height: 75


ListModel {
    id: fruitModel

    ListElement {
        name: "Apple"
        cost: 2.45
    }
    ListElement {
        name: "Orange"
        cost: 3.25
    }
    ListElement {
        name: "Banana"
        cost: 1.95
    }
}
 TableView {
     anchors.fill: parent
```

Modern SPA web development

- HTML 5 + CSS + Javascript in terms for the browser
- Breakpoint/Grid based organisation of CSS styled HTML based items in reaction of
  - Resolution/DPI/Aspect ratio
  - Light/Dark Theme
  - UI changes not by HTTP get next site load but async AJAX calls
- Javascript and CSS linting, transpiling, compiling, compressing


Company name
Features
Enterprise
Support
Pricing
Sign up
Pricing
Quickly build an effective pricing table for your potential customers with this Bootstrap example. It's built with default Bootstrap components and utilities with little customization.
Free
$0 / mo
10 users included
2 GB of storage
Email support
Help center access
Sign up for free
Pro
$15 / mo
20 users included
10 GB of storage
Priority email support
Help center access
Get started
Enterprise
$29 / mo
30 users included
15 GB of storage
Phone and email support
Help center access
Contact us


CONNECT A LOCATION
Desktop
Documents
Downloads
Music
Pictures
Videos
Demo
Chronique
Create File / Note
File Name
new note [20190110~151225]
Enter the content of your file / note
Text (txt)
Markdown (md)
Rich Text (html)
Path
/home/testfiles/Demo/Thumbnail
CANCEL
OK
sample
URL
3 years ago 218 B
sample
DOCX
a month ago 352.52 KB


SEARCH
indexed 20 entries
Search current location
Must contain all of these tags
medium
At least one of these tags
next
None of these tags
Tags
File type
Any type
SEARCH
RESET
TXT
TextFile_002
medium
0 B - 6 days ago
TextFile_003
medium
article
maybe
0 B - 6 days ago
TextFile_004
medium
5star
0 B - 6 days ago
TextFile_005
medium
low
next
0 B - 6 days ago
TextFile_006
medium
next
high
5star
0 B - 6 days ago
TextFile_007
medium
next
5star
0 B - 6 days ago
TextFile_015
medium
article
maybe

Serverless

- Example AWS lambda:
  - Native only supported by native node modules
  - Start and end is a javascript function triggered by e.g. a HTTP call

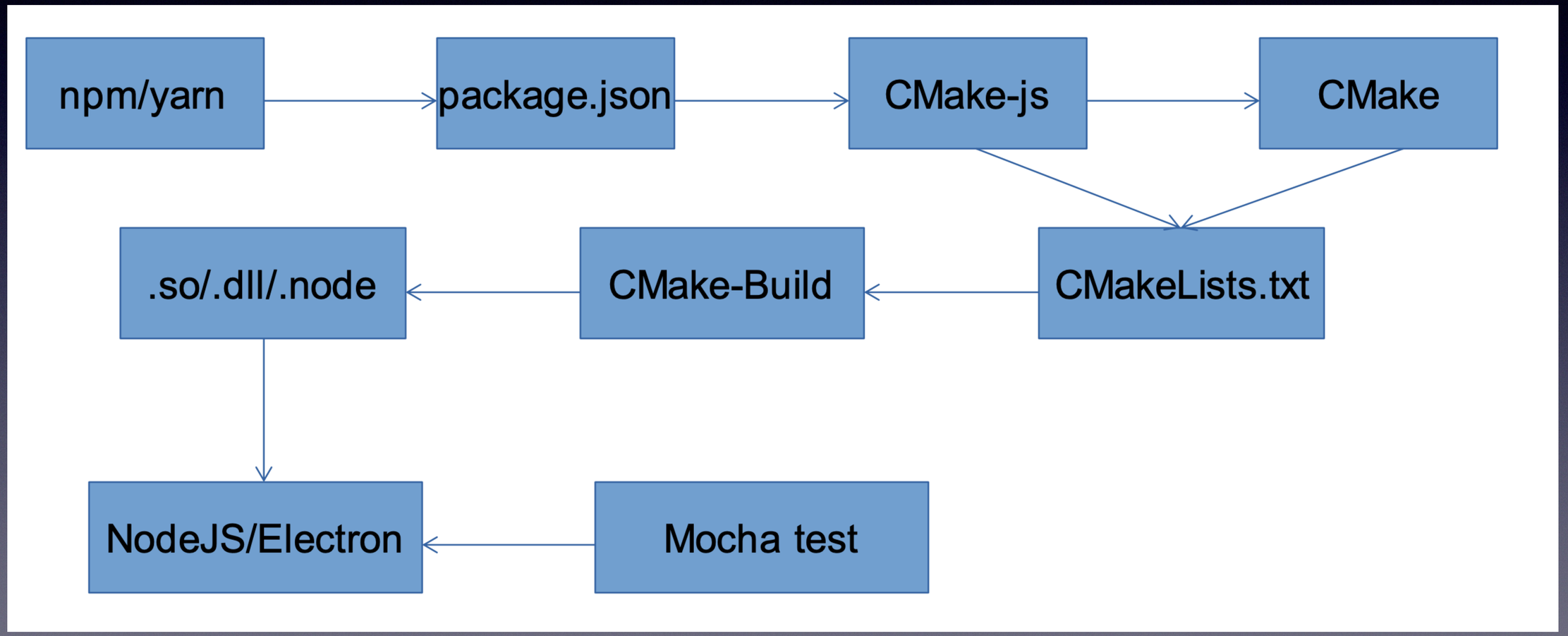
Goals out of C++

- Non C++ only presentation and maybe for orchestration („piping things together“ code)
- Use of CMake, not gyp or vendor supplied or LGPL binaries
- Javascript agnostic ASAP
- No restrictions on lifetime, concurrency patterns

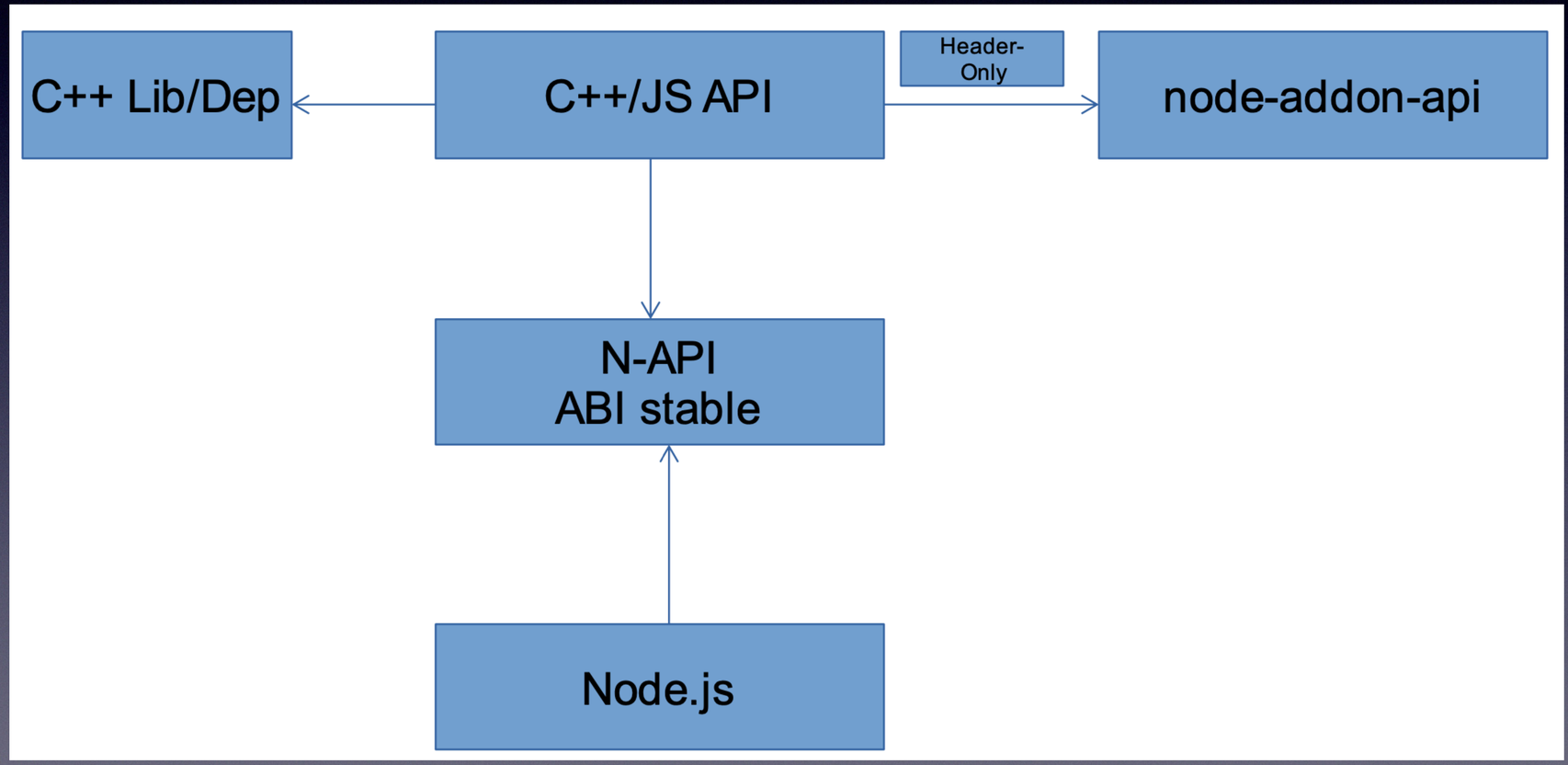
Project overview

- Javascript is interpreted and carbage collected
- Electron is a bundle i.e. framework
  - Bundles Google Chromium for rendering frontend JS/HTML
  - NodeJS for native node modules and app logic
  - Supplies an API for
    - Window management
    - Native system operations like file handling
- https://github.com/Superlokkus/electron_gui

Build


npm/yarn
package.json
CMake-js
CMake
.so/.dll/.node
CMake-Build
CMakeLists.txt
NodeJS/Electron
Mocha test

NAPI


C++ Lib/Dep
C++/JS API
Header-
Only
node-addon-api
N-API
ABI stable
Node.js

Things to know

- Native node module = shared library with .node suffix
- Will be loaded into nodejs i.e. electron executable owned nodejs process
- Javascript code is single threaded GUI-thread schema

Demo and Browse

Dependency management

- Proposed and used structure is
- Parent project with all front end and electron related assets
- One middleware project offering NAPI wrappers as bridge from JS to C++
- Native non NPM/JS aware modules as git submodules in middleware

Learned pitfalls

- Node or electrons DLL handling causes problems with pre main aka dynamic initialization problems -> never use statics or complex constants, but static local variables
- Missing headers on different implementations
- Exception handling to JS is key

C++ outlook

- Shows the need for modern async patterns from boost::asio shared from this pattersn to C++20 coroutines
- std::embed would make so many things possible, e.g. JSON schema compiler code generation
- Redux i.e. flux state model for UI, top state programming

Thank you and Q&A