Cross-platform native GUIs

{trade,pay}offs, {integra,distribu}tion

Zakariyya Mughal

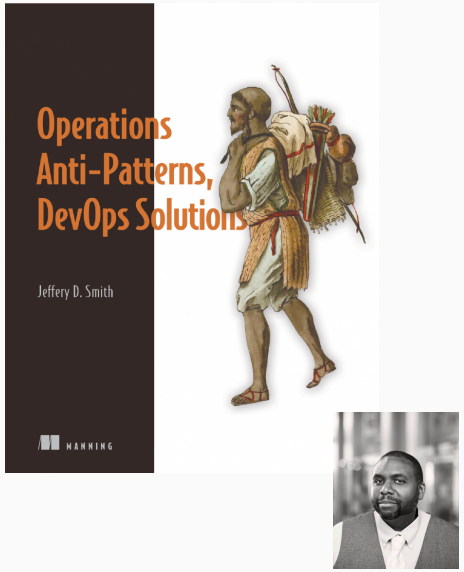
2021-06-10

The Perl and Raku Conference (In the Cloud) 2021

https://github.com/zmughal-biblio/talk-tprc2021cic-cross-platform-native-guis-20210610

Motivation

Operations Anti-Patterns, DevOps Solutions by Jeffery D. Smith



Working in Public by Nadia Eghbal

The competition

Put on developer hat

Web applications are great

- Easy deployment → Easy update
  for developers for users
- Flexible declarative languages for *separation of content and presentation*.
  - HTML, CSS, MathML, SVG

But there are drawbacks

- Web browsers are a sandbox for a reason.

But there are drawbacks

- Web browsers are a sandbox for a reason.
- Platform + browser combination bugs.

But there are drawbacks

- Web browsers are a sandbox for a reason.
- Platform + browser combination bugs.
- Polyfills/shims are... dirty

I want to do X, Y, & Z

- Lots of useful native code already exists that would need to reimplemented.

I want to do X, Y, & Z

- Lots of useful native code already exists that would need to reimplemented.
    - Yes, we have WebAssembly. And tooling like Emscripten.

I want to do X, Y, & Z

- Lots of useful native code already exists that would need to reimplemented.
    - Yes, we have WebAssembly. And tooling like Emscripten.
    - But package management is incomplete. And packages are not well-tested.

I want to do X, Y, & Z

- Lots of useful native code already exists that would need to reimplemented.
    - Yes, we have WebAssembly. And tooling like Emscripten.
    - But package management is incomplete. And packages are not well-tested.
- Accessing that code through a client-server architecture adds overhead for some applications.

Computing APIs

Native	Web browser
OpenGL, OpenCL, Vulkan 1992 2009 2016	WebGL, ~~WebCL~~, WebGPU 2011 2014 upcoming
OS threads 1960s	Web Workers, Worklet 2008–2010 2015–
daemon process	Service Workers 2014
(TCP) socket 1970–1980	WebSocket 2011
key-value store library	Web storage, ~~WebSQL~~, IndexedDB 2009 2010 2015
machine data types	typed arrays 2011

Fusion?

- **Problem** Limited control over memory, storage, battery, and other resources.

Fusion?

- **Problem** Limited control over memory, storage, battery, and other resources.
- **Solution** Embed a specific browser to provide a webview (Electron, NW.js, Chromium Embedded Framework, Apache Cordova).

Fusion?

- **Problem** Limited control over memory, storage, battery, and other resources.
- **Solution** Embed a specific browser to provide a webview (Electron, NW.js, Chromium Embedded Framework, Apache Cordova).
- Gives a lot of control over how resources are passed into the webview part of the application, e.g., through custom scheme handlers.

Put on user hat

User data goes... somewhere

Underappreciated values

- Privacy

User data goes... somewhere

Underappreciated values

- Privacy
- Data interoperability

User data goes... somewhere

Underappreciated values

- Privacy
- Data interoperability
- Future-proof (backups, format shifting)

User data goes… somewhere

Underappreciated values

- Privacy
- Data interoperability
- Future-proof (backups, format shifting)
- Integration with the desktop

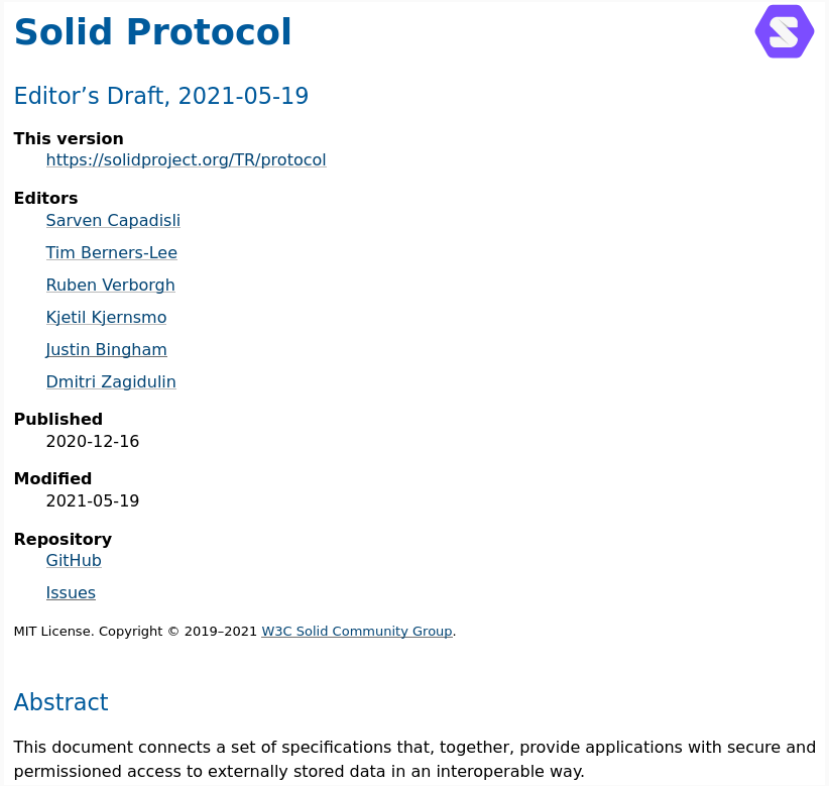
User data goes... somewhere

Data interoperability can provide *separation of data and application*.

User data goes... somewhere

Data interoperability can provide *separation of data and application*.

Possible solution in Solid.



Solid Protocol

Editor's Draft, 2021-05-19

This version
https://solidproject.org/TR/protocol

Editors
Sarven Capadisli
Tim Berners-Lee
Ruben Verborgh
Kjetil Kjernsmo
Justin Bingham
Dmitri Zagidulin

Published
2020-12-16

Modified
2021-05-19

Repository
GitHub
Issues

Abstract

This document connects a set of specifications that, together, provide applications with secure and permissioned access to externally stored data in an interoperable way.

The Lost City of Interaction Design

Integration between applications is limited or sometimes broken:

- Clipboard
- Drag-and-drop

The Lost City of Interaction Design

Widgets are often very non-desktop like:

- Frameworks exist, e.g., qooxdoo

The Lost City of Interaction Design

Widgets are often very non-desktop like:

- Frameworks exist, e.g., qooxdoo
- Not every widget supports:
  - Jumping to items / incremental search
  - Multi-level undo
  - Binding all keyboard shortcuts

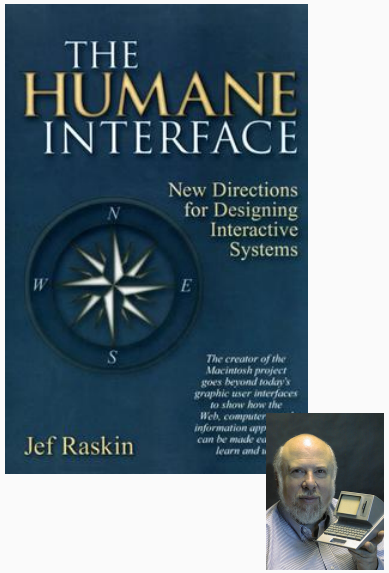
Put on carbon-based lifeform hat

- Visual system was made for object recognition, not reading.
- Latency:

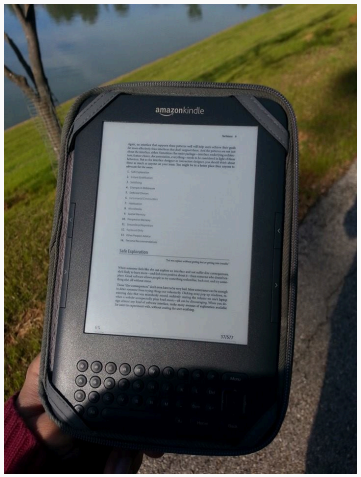
  input → render

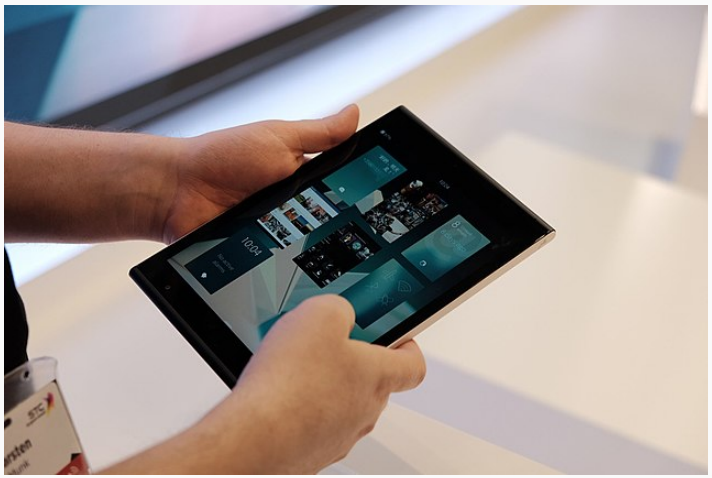
  feedback loop

The Humane Interface by Jef Raskin



Hardware


amazonkindle



Intermission

C shanty

There once was a lib written in C

Installing it was not easy

C shanty

There once was a lib written in C
Installing it was not easy

Tried t'build on Win and go
But linker failed, Bill said no

C shanty

There once was a lib written in C
Installing it was not easy

Tried t'build on Win and go
But linker failed, Bill said no

Soon may the QA come
But not until all checks are done

C shanty

There once was a lib written in C
Installing it was not easy

Tried t'build on Win and go
But linker failed, Bill said no

Soon may the QA come
But not until all checks are done

The OS does segfault throw
The buffer did overflow

The technical part

Solutions on CPAN *today*

Cross-platform:

- IUP
- Prima
- Fl
- Tk
- Tkx
- Wx
- Gtk3

Other:

- Win32::GUI (non-cross-platform), Gtk2 (old)

Gtk3

- Cross-platform

Gtk3

- Cross-platform
- Many language bindings provided through GObject Introspection.

Gtk3

- Cross-platform
- Many language bindings provided through GObject Introspection.
- Glade interface builder

Gtk3

- Cross-platform
- Many language bindings provided through GObject Introspection.
- Glade interface builder
- Interactive debugger

- Cross-platform
- Many language bindings provided through GObject Introspection.
- Glade interface builder
- Interactive debugger

```
export GTK_DEBUG='interactive'
my-gtk-application
```

Gtk3 on *

https://github.com/orbital-transfer-example/perl-gtk3-starter-basic

Gtk3 on Linux

- Use system package manager for native packages

Gtk3 on Linux

- Use system package manager for native packages
- “It just works.”

Gtk3 on Linux

- Use system package manager for native packages
- “It just works.” — me (ca. just now)

Gtk3 on Linux

- Use system package manager for native packages
- “It just works.” — me (ca. just now)
- Docker: some tools don’t like being run as root

Gtk3 on macOS

- macOS has several package managers: tested with Homebrew

Gtk3 on macOS

- macOS has several package managers: tested with Homebrew
- Do not use system Perl (good advice for any platform)

Gtk3 on macOS

- macOS has several package managers: tested with Homebrew
- Do not use system Perl (good advice for any platform)
- Architecture: `x86_64`, will need testing on `arm64`: https://doesitarm.com/

Gtk3 on MSWin32

- Use MSYS2.

Gtk3 on MSWin32

- Use MSYS2.
- `ExtUtils::MakeMaker` hacks

Gtk3 on MSWin32

- Use MSYS2.
- `ExtUtils::MakeMaker` hacks
- `#define MAX_PATH 260`

Gtk3 on MSWin32

- Use MSYS2.
- `ExtUtils::MakeMaker` hacks
- `#define MAX_PATH 260`
- Disable layered windows

Gtk3 on MSWin32

- Use MSYS2.
- `ExtUtils::MakeMaker` hacks
- `#define MAX_PATH 260`
- Disable layered windows

```
BEGIN {
  if( $^O eq 'MSWin32' ) {
    $ENV{GDK_WIN32_LAYERED} = 0;
  }
}

use Gtk3 -init;
```

Building on GitHub Actions

Using Vagrant for local debugging

Provision a VM locally

```
vagrant up buster64 # Debian
vagrant up win10    # Windows 10
#vagrant up macOS   # macOS
```

Creating distributable packages

- Windows: `pacman -Ql`, `PAR::Packer`, WiX Toolset

Creating distributable packages

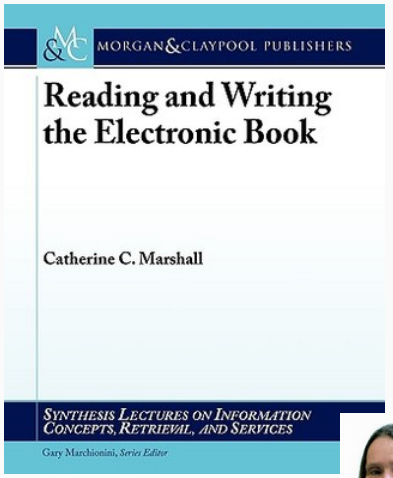
- Windows: `pacman -Ql`, `PAR::Packer`, WiX Toolset
- macOS: Homebrew tap, create-dmg (*TODO*)

Creating distributable packages

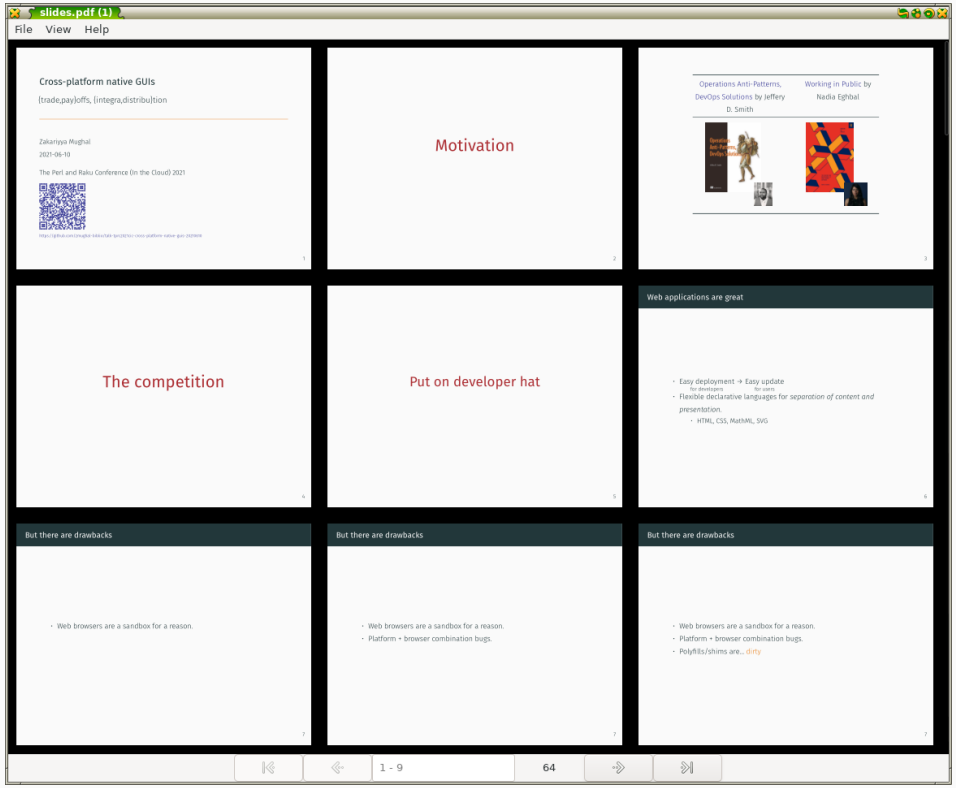
- Windows: `pacman -Ql`, `PAR::Packer`, WiX Toolset
- macOS: Homebrew tap, create-dmg (*TODO*)
- Linux: .deb/.rpm, Flatpak (*TODO*)

Why I wrote a ++(document reader)

Reading and Writing the Electronic Book by Catherine C. Marshall



Curie




```
cpanm -n Renard::Curie
# https://github.com/project-renard/curie
```

Curie stack

- Perl

Curie stack

- Perl
- Gtk3

Curie stack

- Perl
- Gtk3
- Cairo

Curie stack

- Perl
- Gtk3
- Cairo
- MuPDF (Alien::MuPDF)

Curie stack

- Perl
- Gtk3
- Cairo
- MuPDF (Alien::MuPDF)
- Custom scene graph

Curie stack

- Perl
- Gtk3
- Cairo
- MuPDF (Alien::MuPDF)
- Custom scene graph
- Graphene (Alien::Graphene)

Curie stack

- Perl
- Gtk3
- Cairo
- MuPDF (Alien::MuPDF)
- Custom scene graph
- Graphene (Alien::Graphene)
- Kiwisolver (Alien::Kiwisolver)

Curie stack

- Perl
- Gtk3
- Cairo
- MuPDF (Alien::MuPDF)
- Custom scene graph
- Graphene (Alien::Graphene)
- Kiwisolver (Alien::Kiwisolver)
- Festival

Curie stack

- Perl
- Gtk3
- Cairo
- MuPDF (Alien::MuPDF)
- Custom scene graph
- Graphene (Alien::Graphene)
- Kiwisolver (Alien::Kiwisolver)
- Festival
- IO::Async

Future

- AndroWish

Future

- AndroWish
- Port to e-ink device

Future

- AndroWish
- Port to e-ink device
- Binary Perl dist packaging on CI?

Acknowledgements

- Chirag Ghanshani, Stanislav Yotov, Jesus Hernandez
- CPAN Testers, Slaven Rezić, Thibault Duponchelle
- PerlAlien, #native, GTK-Perl mailing list

Questions & Contact

- on IRC: `sivoais` on irc://irc.perl.org/#native (Alien and FFI!) or irc://irc.perl.org/#pdl (scientific and numerical computing!),
- on Twitter: @zmughal,
- on GitHub: https://github.com/zmughal.