@quasar/tauri - v1.0.0-alpha.0

https://github.com/quasarframework/tauri

TAURI

Create tiny, fast and secure cross-platform native apps with the ease of VueJS and the power of the Quasar Framework

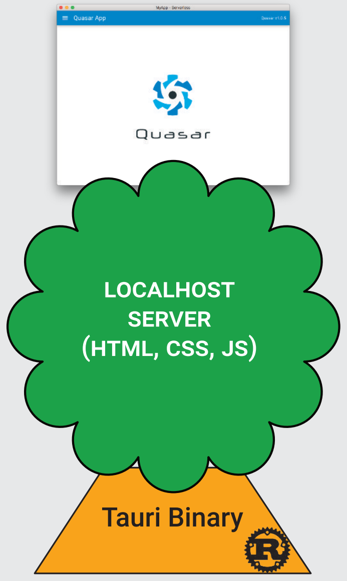
Architectural Design Patterns

All Design Patterns shown here are configurable entirely within quasar.conf.js.

Tauri is a new mode for building Quasar-Framework Apps that creates software binaries for all major desktop platforms. So why should you choose Tauri?

- **BUNDLE SIZE** of a vanilla Tauri app is less than 3 MB - about 140 MB smaller than what you get with Electron.
- **MEMORY FOOTPRINT** is less than half of the size of an Electron app built from the same codebase.
- **SECURITY** is Tauri's biggest priority and we take it so seriously that we innovate to help you keep hackers out of your apps.
- **RELIABILITY** of the underlying code base is why all critical libraries have been forked and will be perpetually maintained.
- **FLOSS** licensing is regretfully impossible with downstream Chromium consumers, like Electron.

Cloudish



The **Cloudish** recipe is a pattern for maximum flexibility and app performance.

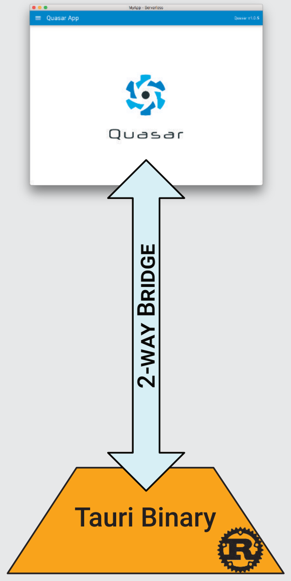
Features:
- ships with a rust-based localhost server
- works out of the box
- can utilize CSP for extra security

Best When:
- starting out building Tauri apps if you have never used Rust before.

Pros:
- easy to use
- best WebAPI compliance

Cons:
- hard to secure against Man in the Machine attacks
- gives you an easy excuse not to learn Rust

Trollbridge



The **Trollbridge** recipe is a pattern for the highest degree of operational security.

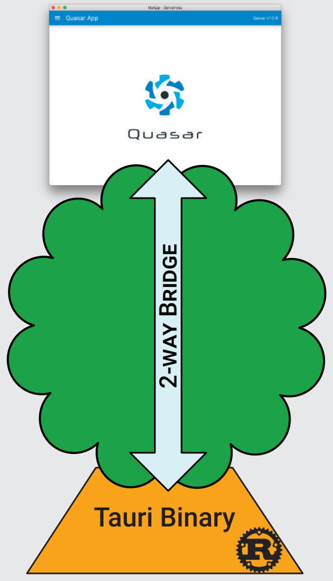
Features:
- render UI securely at bootstrap
- promise based message passing
- RW access to filesystem
- STDOUT access to other binaries
- extensible with Rust functions
- whitelist for functional codegen
- runtime message salting
- fASLR & AoT Compiling

Best When:
- you are paranoid about security but still need the power of Rust.

Pros:
- highly configurable and infinitely extensible

Cons:
- rust skills virtually required

Cloudbridge



The **Cloudbridge** combines the flexibility of a localhost and the security of the bridge.

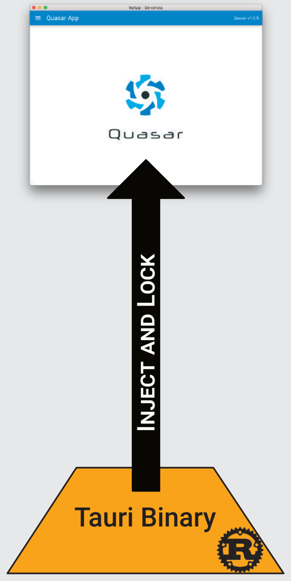
Features:
- all bridge features
- all localhost features

Best When:
- your project is complex and you need all the power available.

Pros:
- best of both worlds

Cons:
- rust skills helpful
- complexity can make it challenging to keep your application from becoming bloated

Hermit



The **Hermit** recipe is a pattern for ultimate application isolation where all logic is self-contained.

Features:
- render UI securely at bootstrap
- no communication with Rust
- CSP blocks all external resources
- fASLR & AoT Compiling

Best When:
- you want to limit interaction with anything outside of the scope of the user interface like for a game.

Pros:
- safest possible distribution target

Cons:
- limited device interaction

@quasar/tauri - v1.0.0-alpha.0

https://github.com/quasarframework/tauri