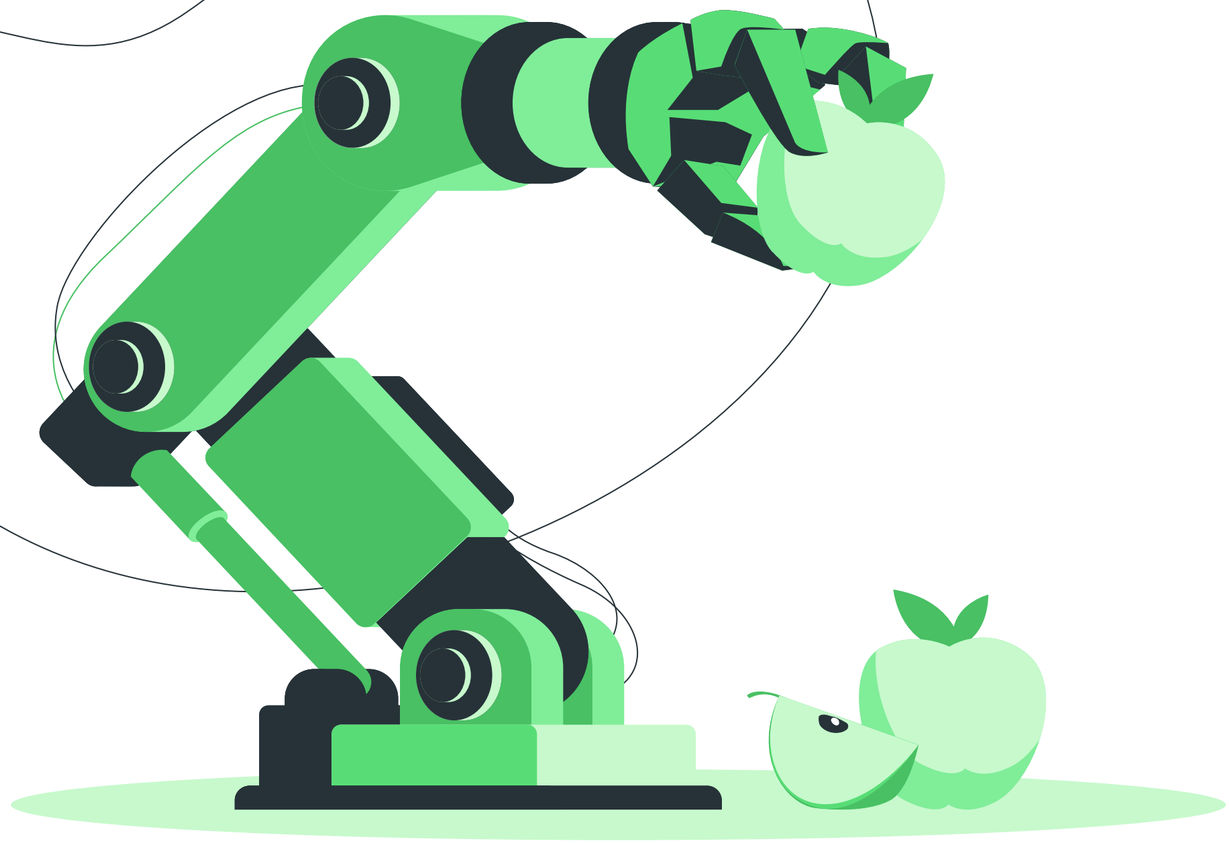


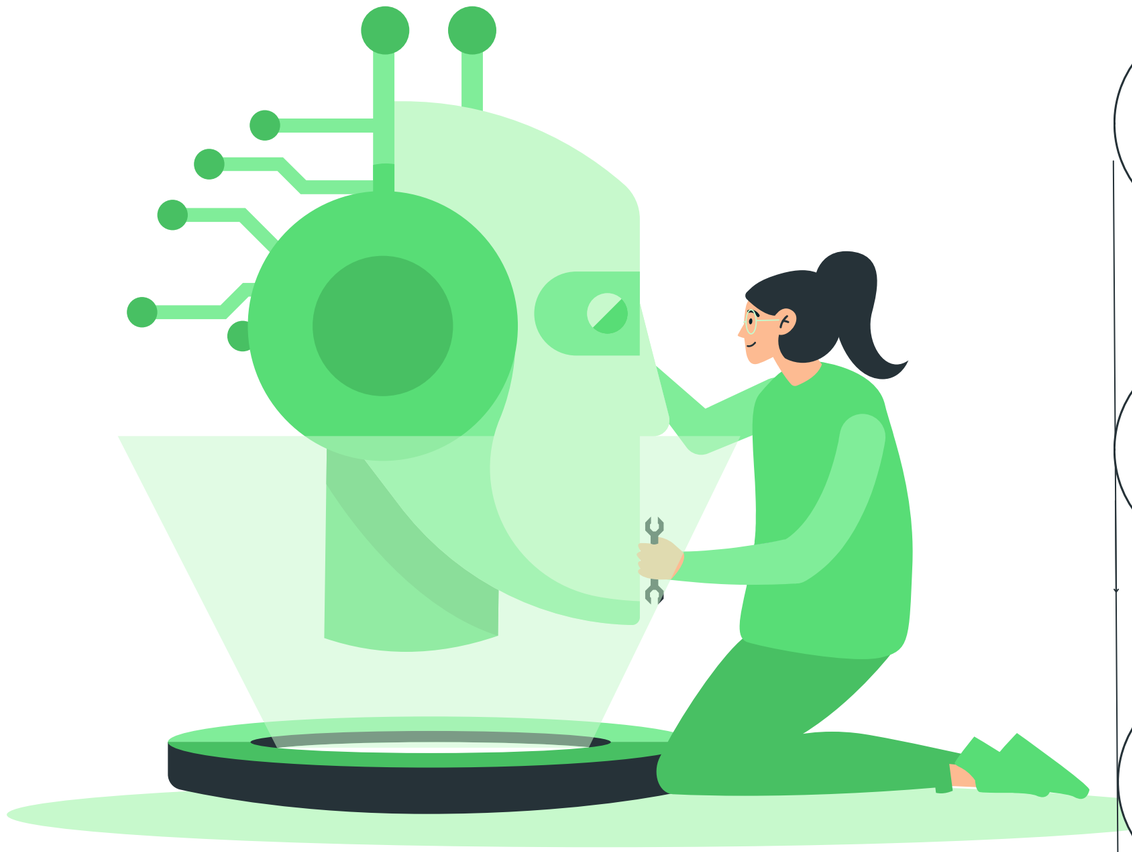
GNUstep Conceptual Architecture

Presented by: Kiarash Mirkamandari and Ebrahim Haghshenas

Agenda

1. Motivation &Context
2. Top-Level Architecture
3. Subsystems & Interactions
4. Use Cases & Sequence Diagrams
5. Derivation Process & Alternatives
5. Lessons Learned & Limitations
5. Conclusion & Future Directions

Motivation & Context

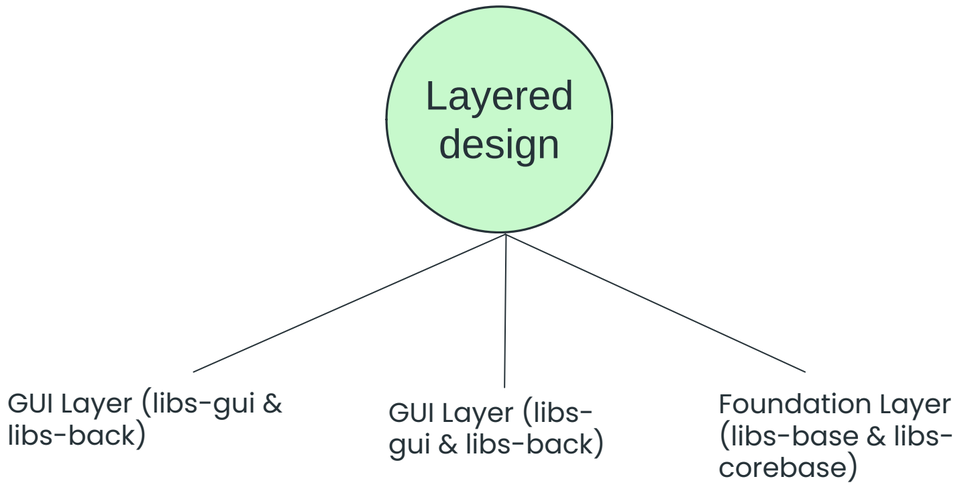


Cross-platform GUI challenge
native look & feel across OSes

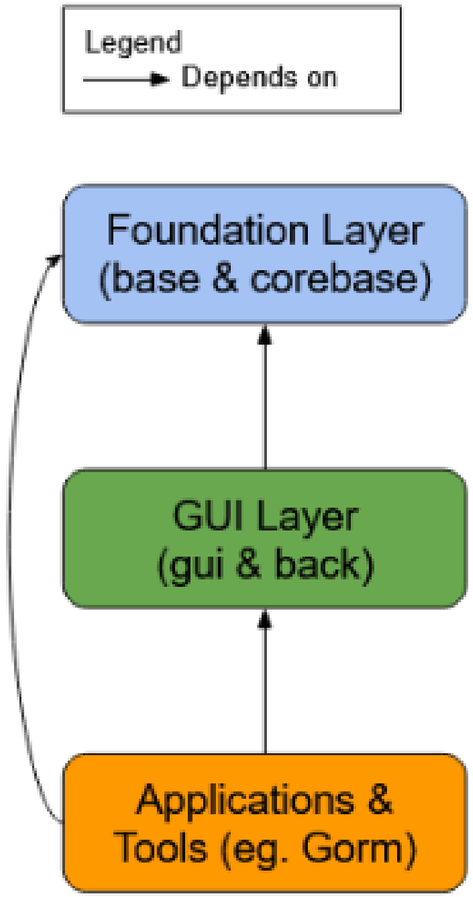
GNUstep reimplements the OpenStep specification
unified development approach

Objective-C foundation
Simplifies portability and code reuse

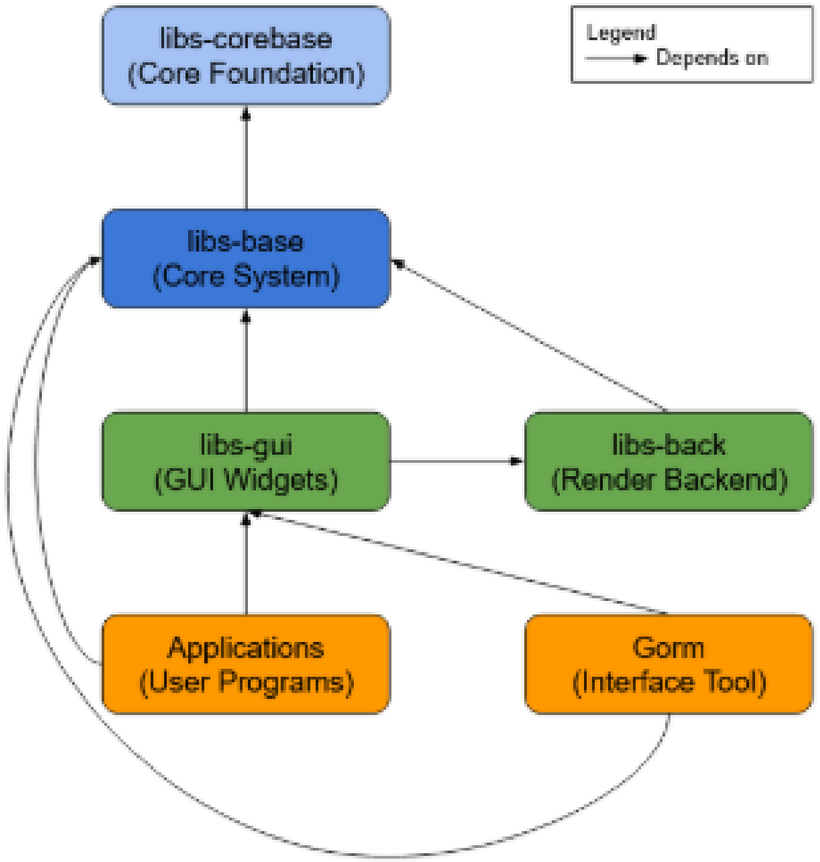
Top-Level Architectural Style



- MVC, object-oriented, event-driven styles
- Layers enable maintainability & portability



Subsystems & Components



libs-base

1

Core classes (strings, arrays, etc.)

2

libs-corebase

Low-level (similar to Core Foundation)

3

libs-gui

GUI widgets (windows, buttons, views)

4

libs-back

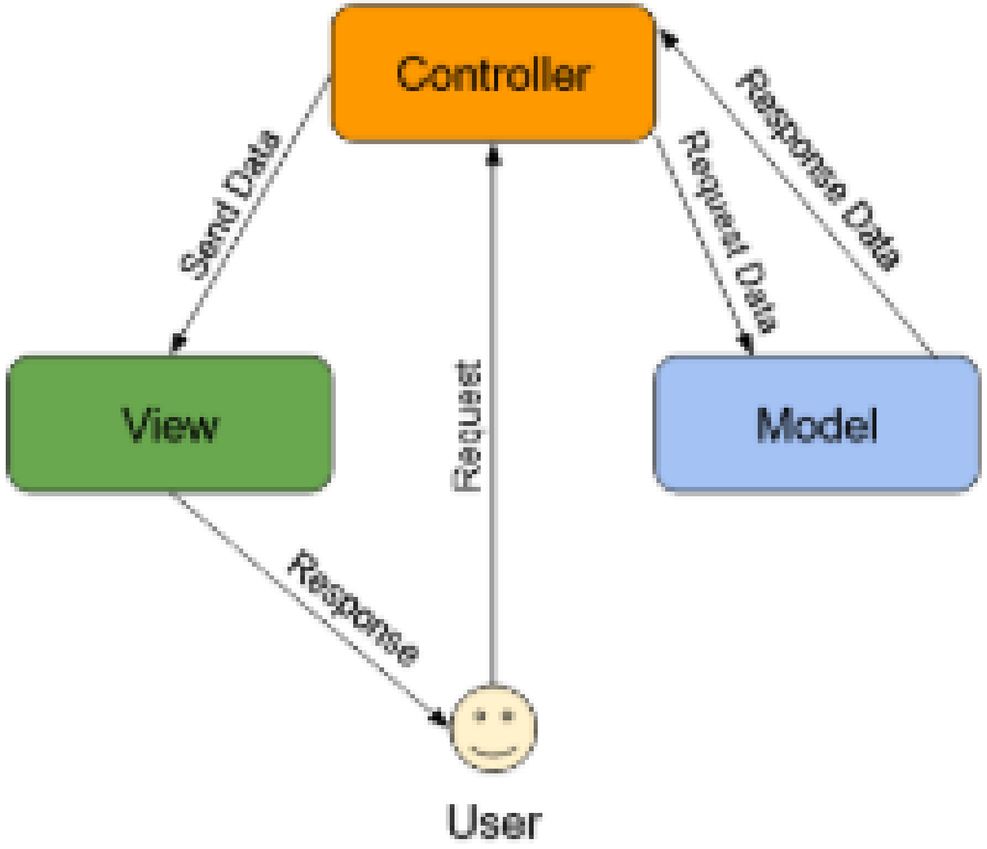
Rendering backend (X11, Win32, etc.)

5

Gorm

GUI builder tool

MVC Emphasis



1. **Model**
   Business logic/data (libs-base)

2. **View**
   UI elements (libs-gui)

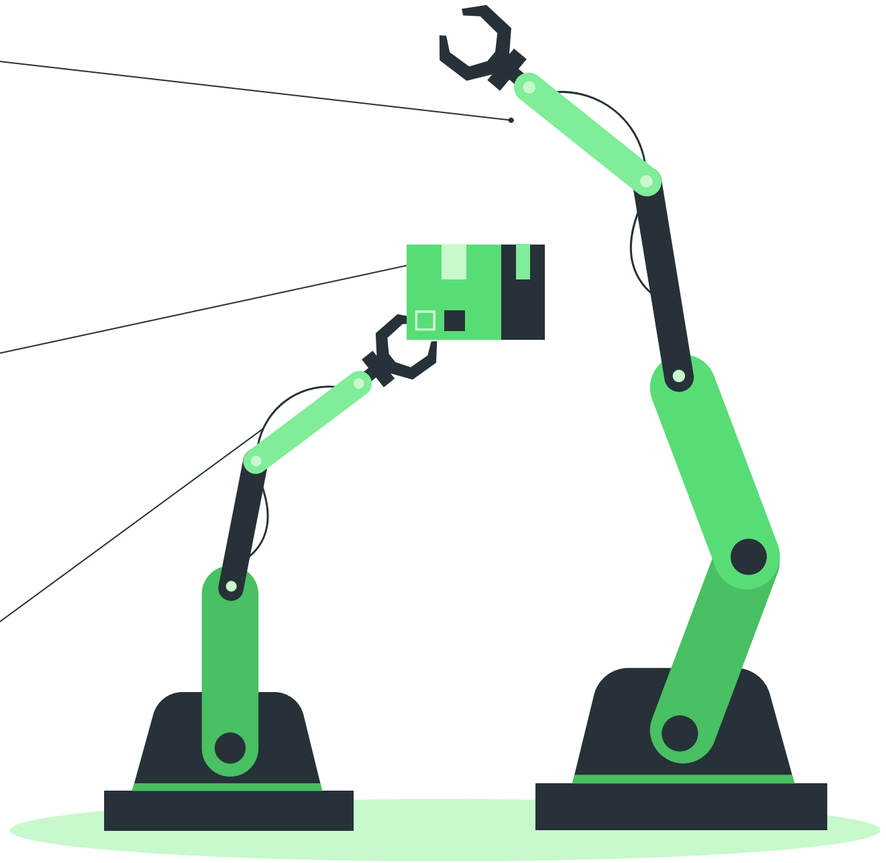
3. **Controller**
   Links user actions to models & views

Concurrency & Data Flow

Event-driven main loop for GUI responsiveness

Multithreading support (POSIX, operation queues)

Synchronization ensures thread-safe rendering



Use Case 1: App Launch

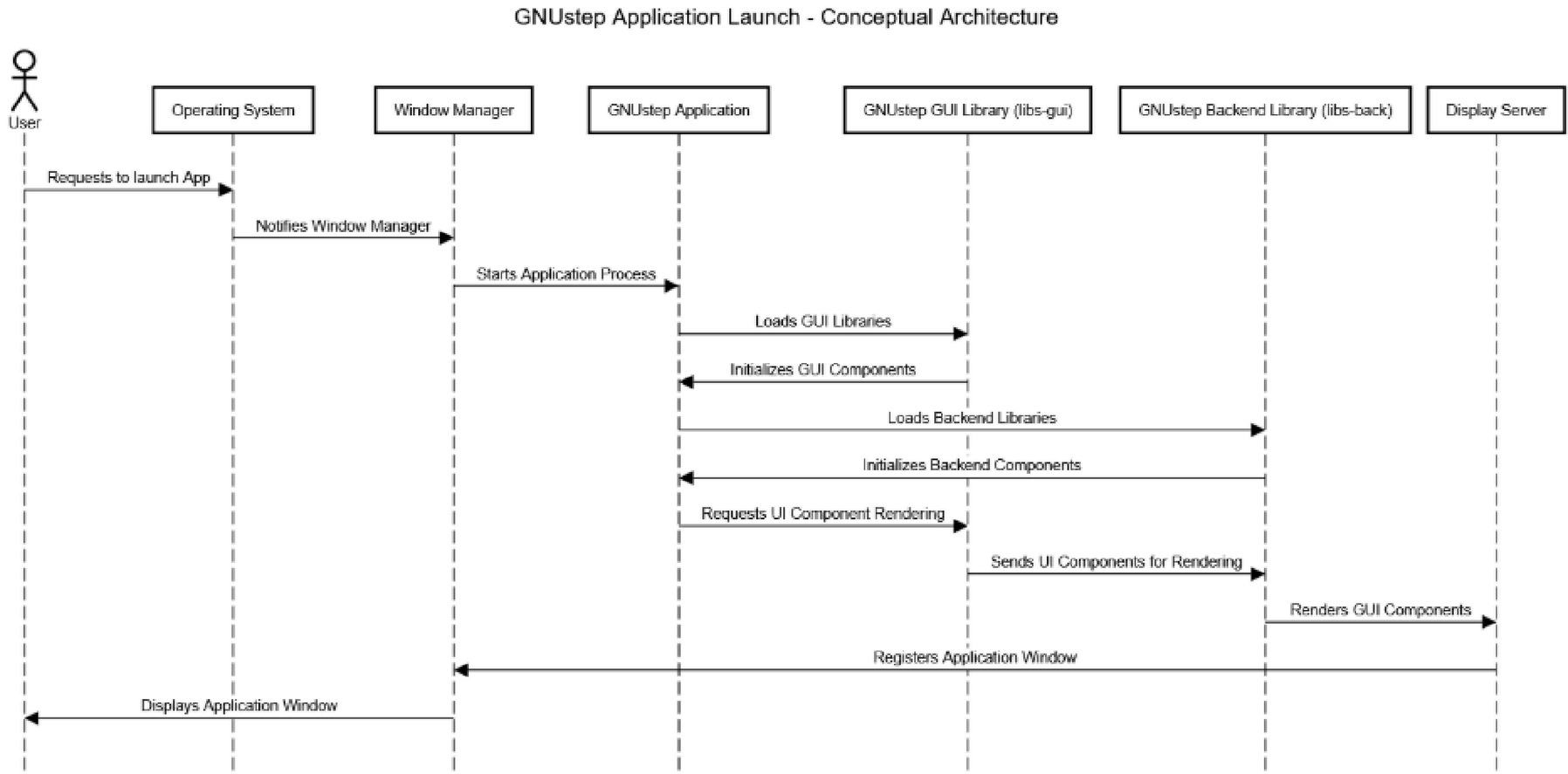
User/OS requests launch → Window Manager spawns process

---

libs-gui loads UI components; libs-back initializes rendering

---

Application window is displayed



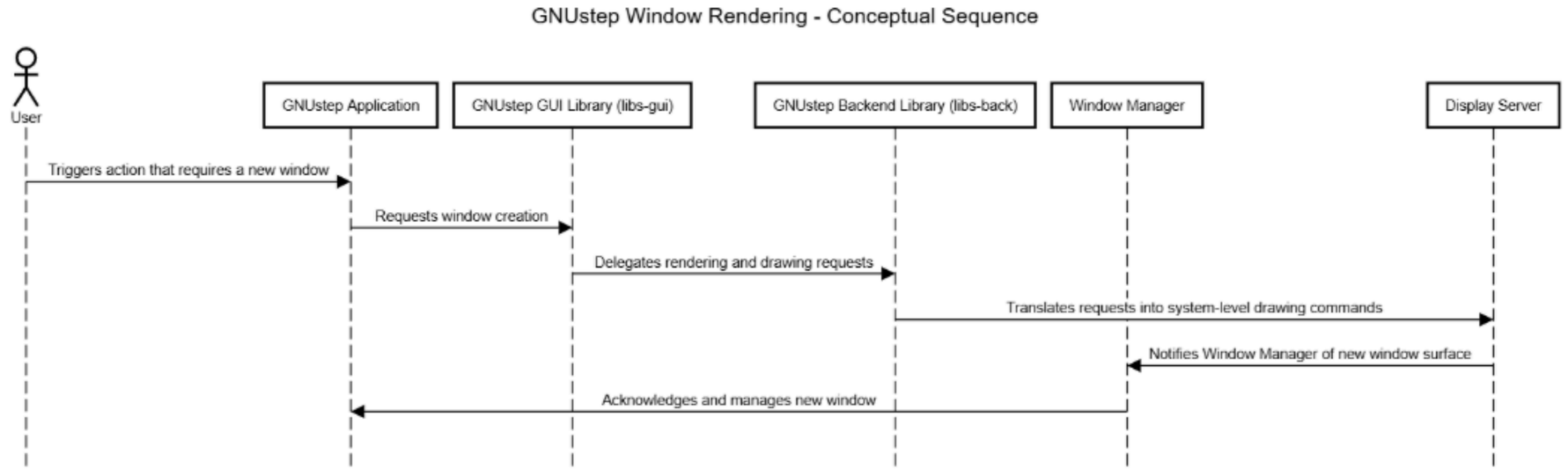
Use Case 2: Rendering a Window

User action →
GNUstep app →
libs-gui →
libs-back

---

libs-back translates to system-level drawing

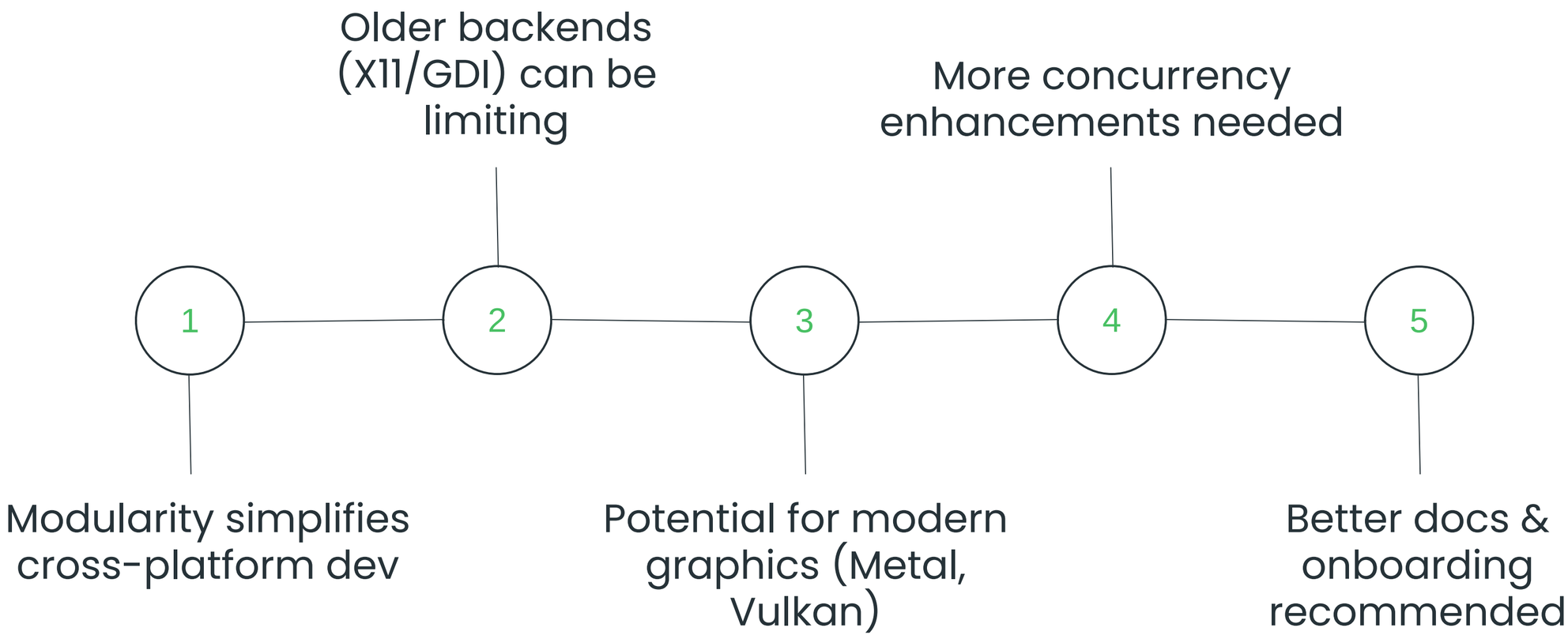
Window Manager notifies creation of new window surface



Derivation & Alternatives

Modular vs. monolithic	Direct OS Calls vs. libs-back	Objective-C vs. C++
• Modular approach chosen for flexibility and maintainability	• libs-back abstraction preserves OS independence	• Objective-C retained for OpenStep compatibility

Lessons & Limitations



Conclusion

GNUstep’s layers + MVC

=

strong portability & structure

Future: advanced concurrency, updated graphics backends