



Conceptual Architecture of GNUStep

GAMERS NEVER GIVE UP

Jawad Ahmed (Leader), Meagan Mann (Presenter), Ethan Nguyen (Presenter), Zoya Zarei-Joorshari,
Kim Hyun Bin, Ripley Visentin

<https://youtu.be/v61Z3Pzd5fo>




INTRODUCTION

Why Frameworks Matter

- Application development requires structured frameworks.
- Apple's Cocoa (from NeXT's OpenStep) is macOS-exclusive.
- Need for a cross-platform alternative.

What is GNUstep?

- Open-source, cross-platform GUI development framework.
 - Based on OpenStep, the predecessor to Cocoa.
 - Written in Objective-C for portability.
- 

Key Functionalities of GNUStep

Objective-C Framework Implementation

GNUstep offers a free software implementation of the Cocoa frameworks. This includes classes for developing both GUI and non-GUI applications.



Cross-Platform Application Development

GNUstep allows the creation of advanced desktop applications that are compatible across different operating systems.



Development Tools and IDE Support :

GNUstep provides tools for command-line and GUI application development.

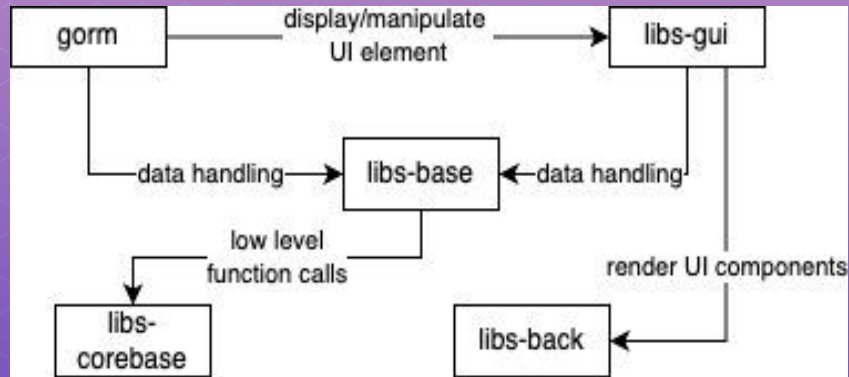


Architectural Style

Object-Oriented Architecture: GNUstep follows an object-oriented style, where multiple components function as autonomous, encapsulated libraries that interact through function calls.

Interacting Components: Instead of a strict layered structure, components communicate dynamically, invoking each other as needed.

Evaluation of Layered Style: While a layered approach was considered, GNUstep's flexible, interconnected design aligns more closely with an object-oriented model.



Main Components



Libs-back

Handles rendering across platforms, translating graphics for different systems.



Libs-base

Provides essential system services and forms the foundation.



Libs-corebase

Adds low-level utilities like memory management and networking.



Libs-GUI

Implements the AppKit API for UI elements.



Gorm

A GUI builder for designing interfaces visually.

Interactions Between Parts

Libs-base

Both **libs-corebase** and **libs-GUI** depend on **libs-base** for core services like object creation and memory management.

Libs-corebase

Extends **libs-base**. It facilitates interaction between the components, supporting the functionality of **libs-GUI** and **libs-back**.

Libs-GUI

It works closely with **libs-back** to render the graphical elements. **libs-GUI** relies on **libs-corebase** for system-level support.

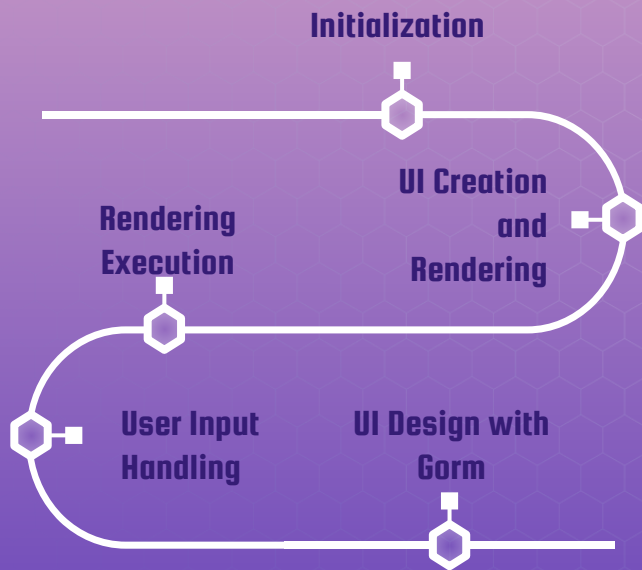
Libs-back

libs-back serves as the GUI backend connecting **libs-GUI** to rendering systems. Translates commands from **libs-GUI** into instructions.

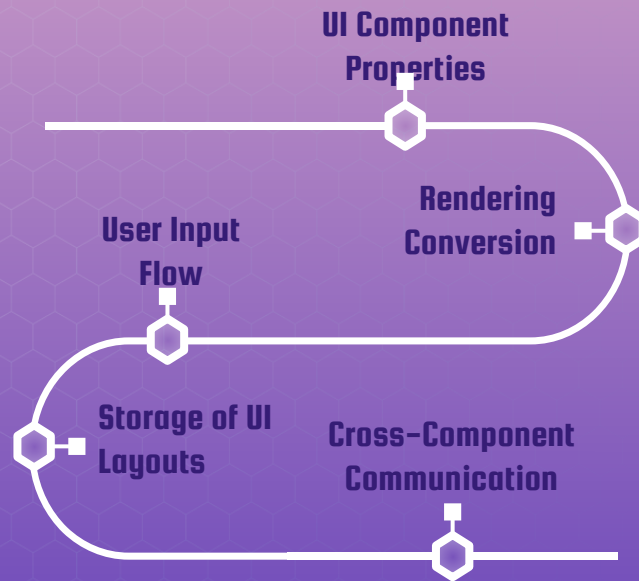
Gorm

Relies on **libs-GUI** to display UI elements and **libs-back** for rendering them across different platforms, ensuring portability of the designs.

Control Flow



Data Flow



Tasks - being split into sub-tasks



Concurrency



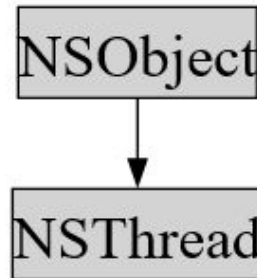
threadPriority

+ (double) **threadPriority**;

Availability: MacOS-X 10.2.0, Base 1.2.0

Return the priority of the current thread.

NSThread : NSObject



Declared in:

Foundation/NSThread.h

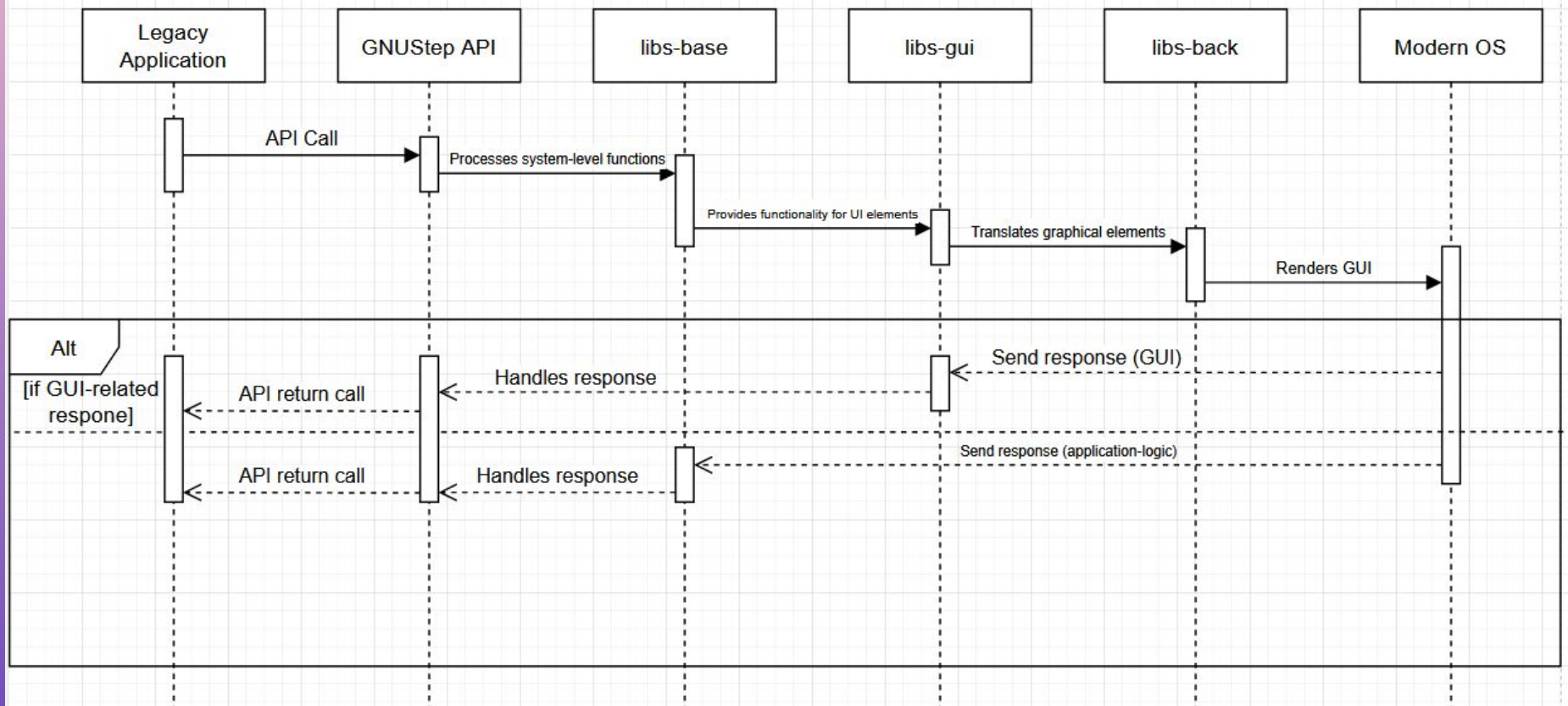


Use Case I: Modernizing Legacy OpenStep Applications

An organization has legacy software originally built on OpenStep. They need to update the application while maintaining compatibility with modern operating systems.

Reviving and Modernizing Classic OpenStep Applications

- Provides an API similar to OpenStep, allowing for easier porting of legacy applications
- Avoids vendor lock-in by transitioning from proprietary Apple/macOS-only dependencies to an open-source environment



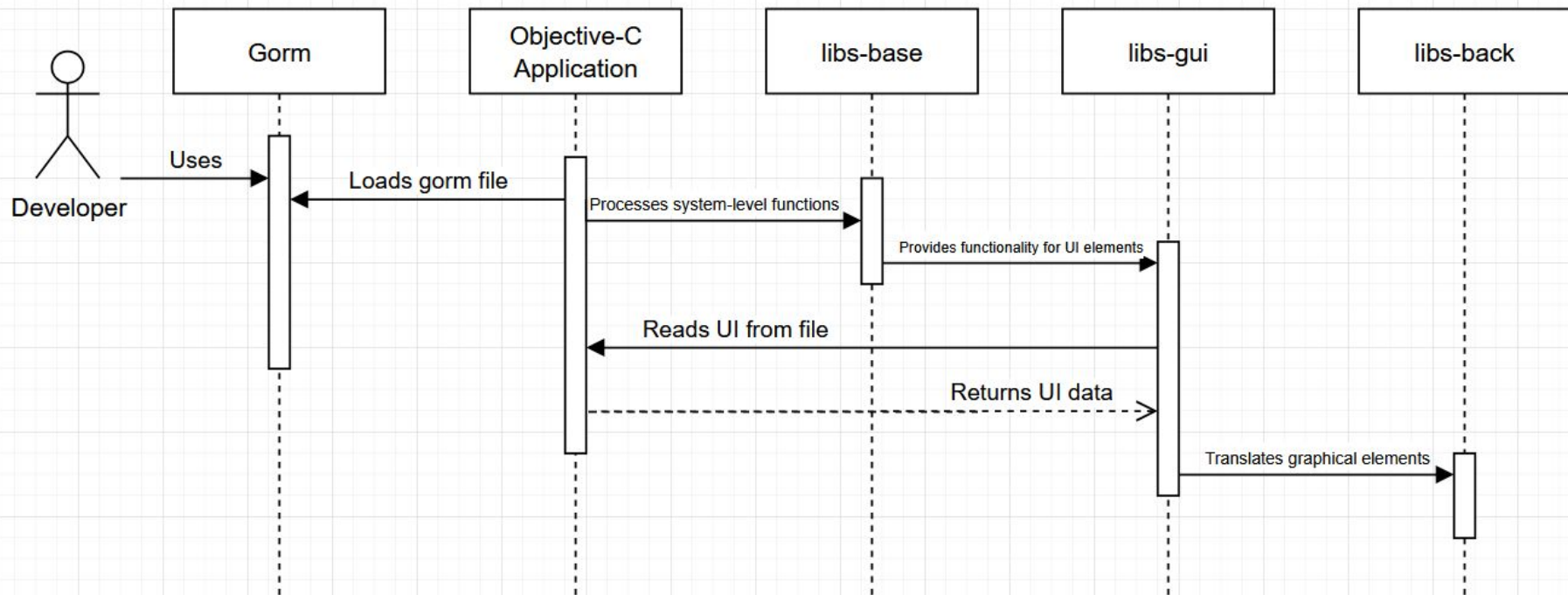


Use Case 2: Cross-Platform Development with Objective-C

A dev team wants to build a cross-platform GUI application in Objective-C but doesn't want to be locked into Apple's ecosystem (Cocoa/macOS). They also want an open-source alternative that works on Linux and Windows.

Building Portable, Open-Source Software in Objective-C

- Allows developers to use Objective-C while targeting multiple operating systems.
- Provides Cocoa-compatible APIs, making it easier for teams familiar with macOS development to transition while also being free and open-source.



System Evolution

libs-base

Key Versions:

1.28.0 - portability improvements and bug fixes, including support for native locking and overlapped I/O on Windows

1.30.0 - weak references became available in portable code

libs-gui

Key Versions:

0.9.3 – Spell checker re-implemented with libaspell, NSComboBox and NSToolbar reworked.

0.11.0 – Keyed encoding added to all GUI classes, improved Nib loading, better theme support.

0.14.0 – Added glyph generation class and old NeXT method for compatibility.

0.16.0 – Stable release with improved Nib loading, document auto-saving, and multiple file extensions.

0.22.0 – Major release with GUI scaling, character panel, color picker, and new image format support.

0.28.0 – Added support for modern XIB files.

gorm

Key Versions:

0.0.1 – Initial release with basic framework, drag-and-drop elements, and window creation.

0.2.0 – Introduced custom class support and improved drag-and-drop, sound, and image handling.

0.4.0 – Introduced menu inspectors and the ability to create custom subclasses.

0.7.5 – Added reparenting feature for changing class hierarchy and NSFontManager support.

0.9.0 – Enabled image and sound drag-and-drop, improved formatter inspectors, and added method support.

0.9.2 – Latest release, primarily bug fixes.

System Evolution

libs-back

Key Versions:

0.0.1 - Introduced as the GNUstep 'back' library, providing a backend component for the GNUstep GUI Library. It serves as an interface to various windowing systems, enabling the GUI library to function across different platforms.

0.9.4 - Implemented significant improvements in rendering performance and added support for additional display servers.

0.10.0 - Enhanced compatibility with newer windowing systems and included updates for better font handling.

0.11.0 - Introduced support for advanced graphics features and improved integration with the GNUstep GUI Library.

0.12.0 - Focused on bug fixes and stability improvements, ensuring reliable performance across supported platforms.

libs-corebase

Key Versions:

Initial Release: Introduced as the GNUstep CoreBase Library, providing general-purpose, non-graphical C objects.

Subsequent Updates: Focused on enhancing stability and expanding functionality to align more closely with CoreFoundation features

Conclusion

- Object-Oriented Architectural Style
- 5 Major Components: libs-base, libs-corebase, libs-gui, libs-back, gorm
- Future Proposals?

```
** Message: Unable to use GNOME Shell's builtin screenshot interface, resorting to fallback X11.
```


Lessons Learned

- Importance of Documentation
- Older Project
 - Many versions/contributors
 - Predecessors + influences
- Communication on Task Division
- Teamwork





Thanks!
Questions?