Group 10: GNU step conceptual architecture presentation

VIDEO URL

CISC322 A1 Group 10 Presentation

GROUP 10

Members

- Chen, Henry – report creation
- Chiepe, Lore – presenter and slide creation
- Jokhu, Ryan – report creation
- Stephens, Zachary – report creation
- Uwefoh, Chuka – Group Lead
- Wang, James- presenter and slide creation

AGENDA

Purpose: Describe the architecture of GNUstep.

- Introduction to GNUstep
- Conceptual Architecture Overview
- Component Interactions
- Workflow of GNUstep
- Evolution of GNUstep
- Concurrency Mechanisms
- Conclusion

WHAT’S GNU STEP?

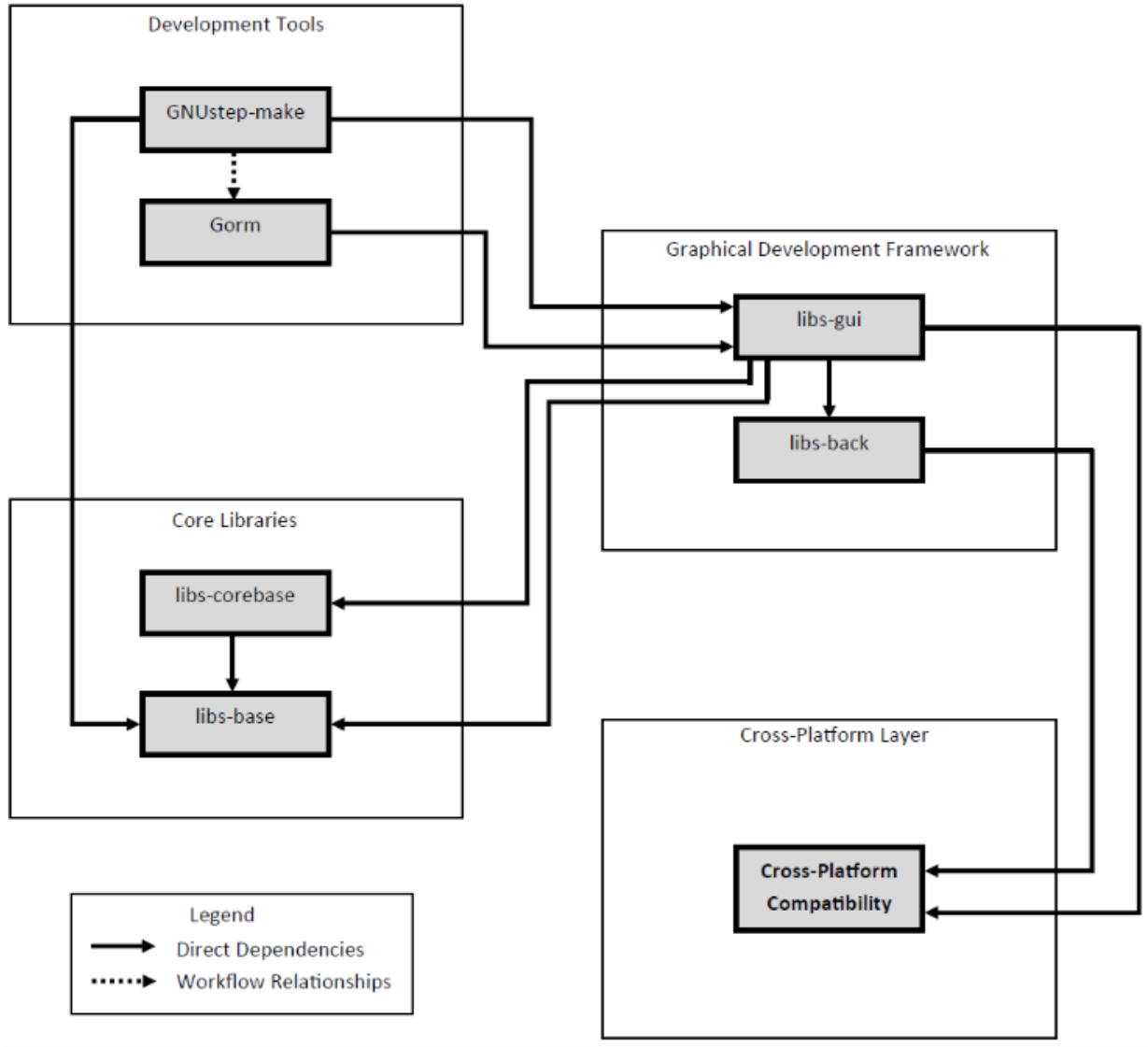
GNU step is an open-source framework for cross-platform application development.

Implements the OpenStep API with an Objective-C runtime.

Provides GUI, backend, and application services.

Used in application development, operating system environments, and research projects.

CONCEPTUAL ARCHITECTURE OVERVIEW



Architectural Style: Layered Architecture.

Key Components:

- lib-Base Layer: Core system utilities (file system, networking, threading).
- lib-GUI Layer: GUI framework for building applications.
- lib-Back Layer: Platform-specific rendering (X11, Wayland, Windows).
- Development Tools: Compilers, debuggers, and build automation tools

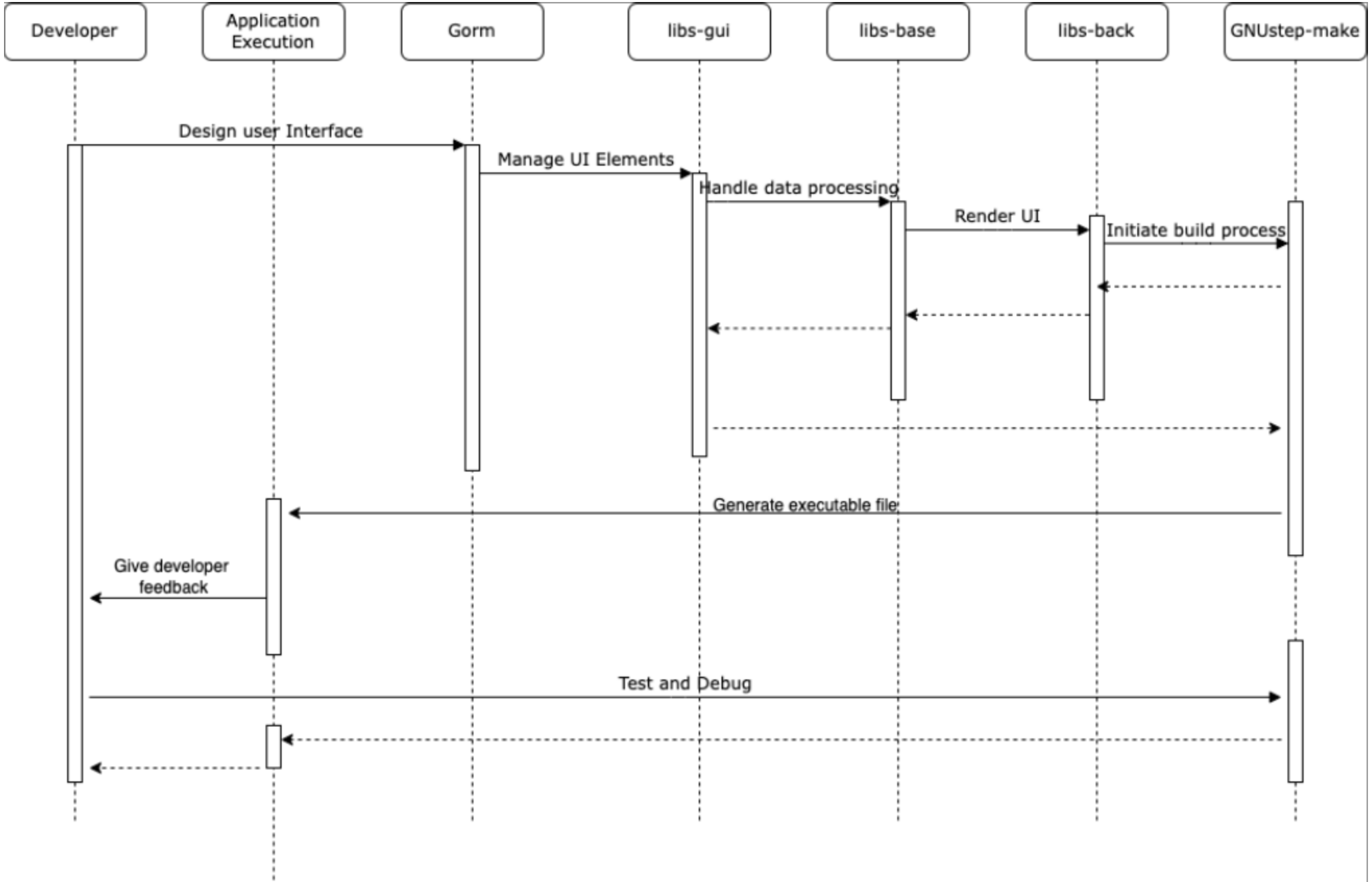
Component Interactions

Cross-Platform Development: The system ensures platform independence by abstracting low-level details such as graphics rendering and windowing systems.

Tools for Development:

- **Gorm**: A GUI design tool that simplifies the creation of graphical user interfaces.
- **GNUstep-make**: The build system for managing dependencies and compiling applications, integrating seamlessly with other components..

Workflow of GNUstep



Evolution of GNUstep

Core Improvements

- Core library refinements
- Enhanced UI/Logic integration
- Modern practice alignment

Developer Tools

- ProjectCenter IDE updates
- Gorm GUI builder updates
- Workflow optimization

Technical Features

- GCD Integration
- Multi-threading support
- Cross-platform compatibility

Community Impact

- Global collaboration
- User feedback & integration
- Regular updates & fixes

Concurrency

Concurrency Implementation

Threading Support

- POSIX Threads
- NSThreads

GCD Integration

- Task queuing
- Parallel processing

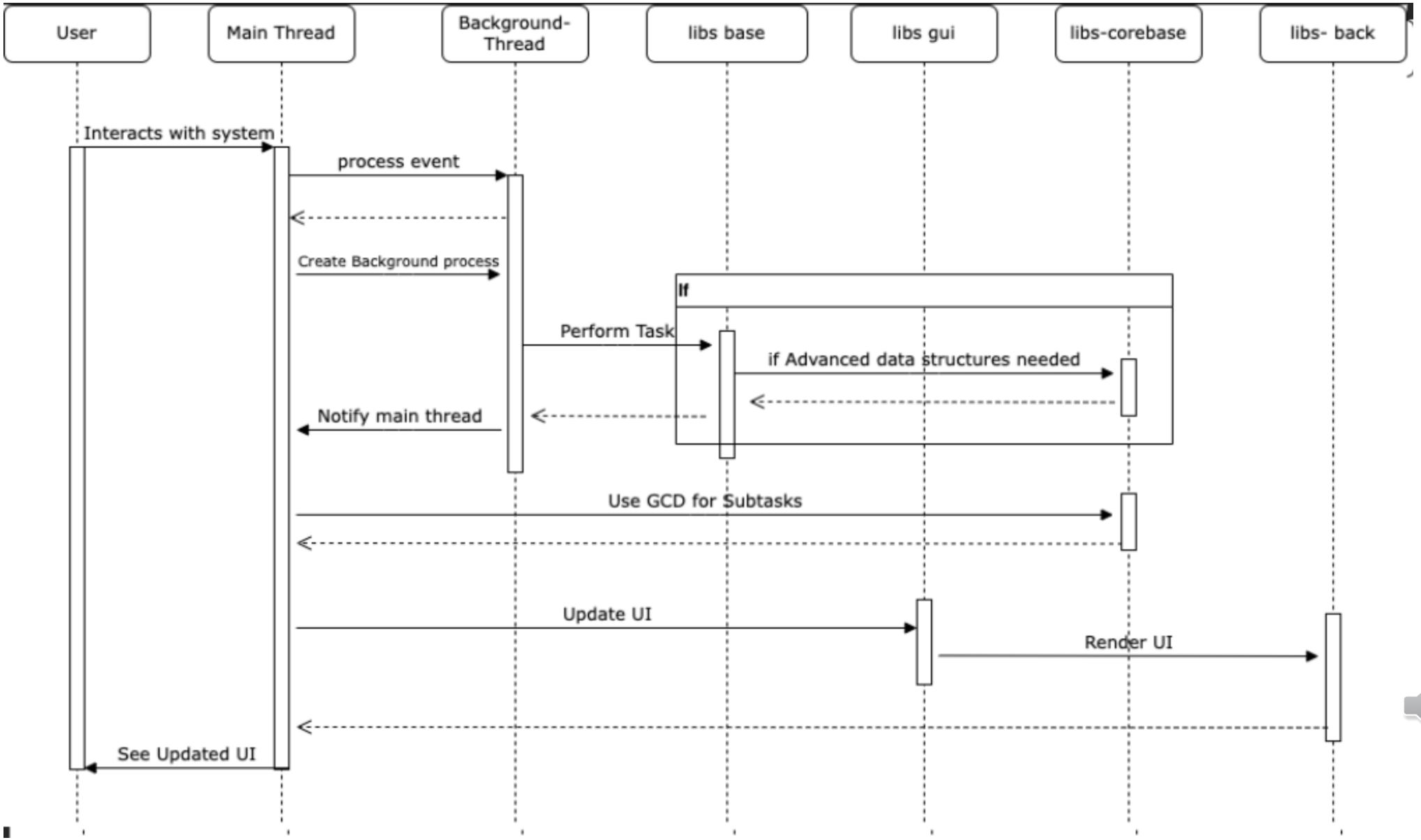
Async Operations

- NSOperations
- Background tasks

Thread Safety

- NSLock
- Resource management

Concurrency – Use Case



SUMMARY

Conclusion

Modular Architecture

- Layered component design
- Flexible integration compatibilities and inter-component communication

Core Structure

- 4 main components
- Optional components

Current status

- In active development
- Latest: Feb 2025

Future Direction

- Enhanced Windows compatibility
- Cross-platform maintenance

References and Links

- GNUstep Official Documentation: https://www.gnustep.org/documentation/
- GNUstep GitHub Repository: https://github.com/gnustep/gnustep
- GNUstep Wiki: https://wiki.gnustep.org
- GNUstep Tutorials: https://www.gnustep.org/resources/tutorials/
- GNUstep Mailing Lists for Community Support: https://www.gnustep.org/resources/mailing-lists/