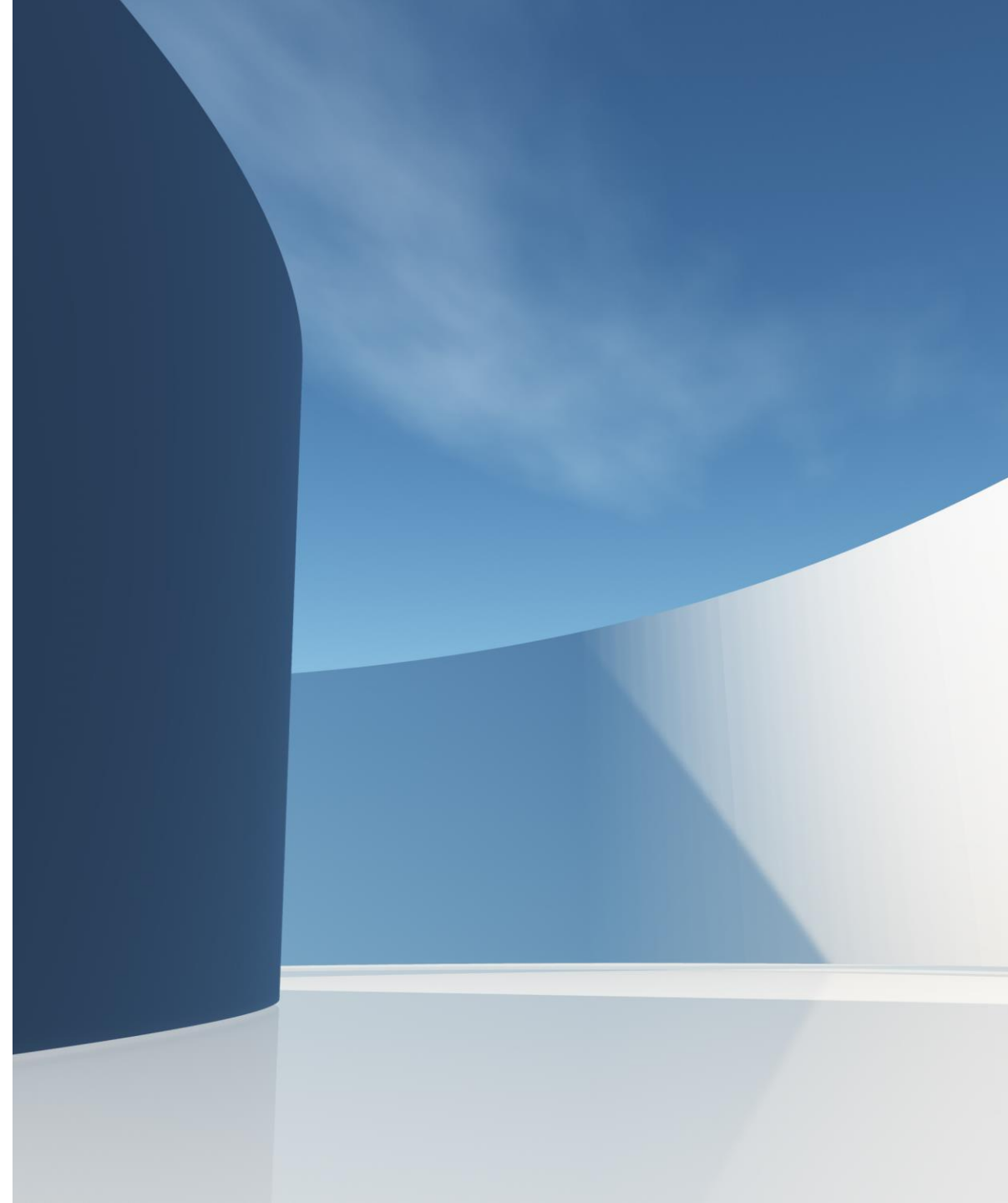

GNUSTEP CONCEPTUAL ARCHITECTURE

Group 12

CISC/CMPE 322/326

Link to video: <https://youtu.be/37RJ6hrGUrE>

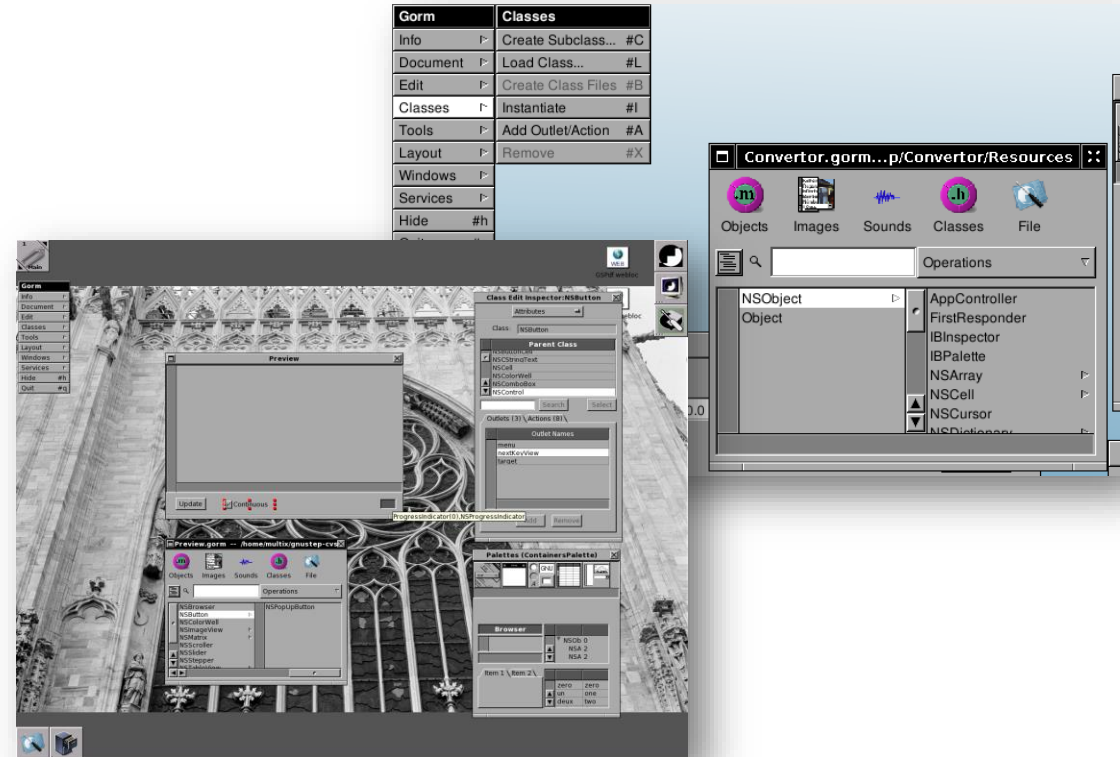


GROUP MEMBERS

- Daniel Tian (Presenter): 21dt41@queensu.ca
 - Samuel Tian (Presenter): 21st114@queensu.ca
 - James Choi: 19jc132@queensu.ca
 - Christian Pierobon: christian.pierobon@queensu.ca
 - Luca Spermezan: 22ls18@queensu.ca
 - Andrew Bissada (Leader): 21ajb37@queensu.ca
-

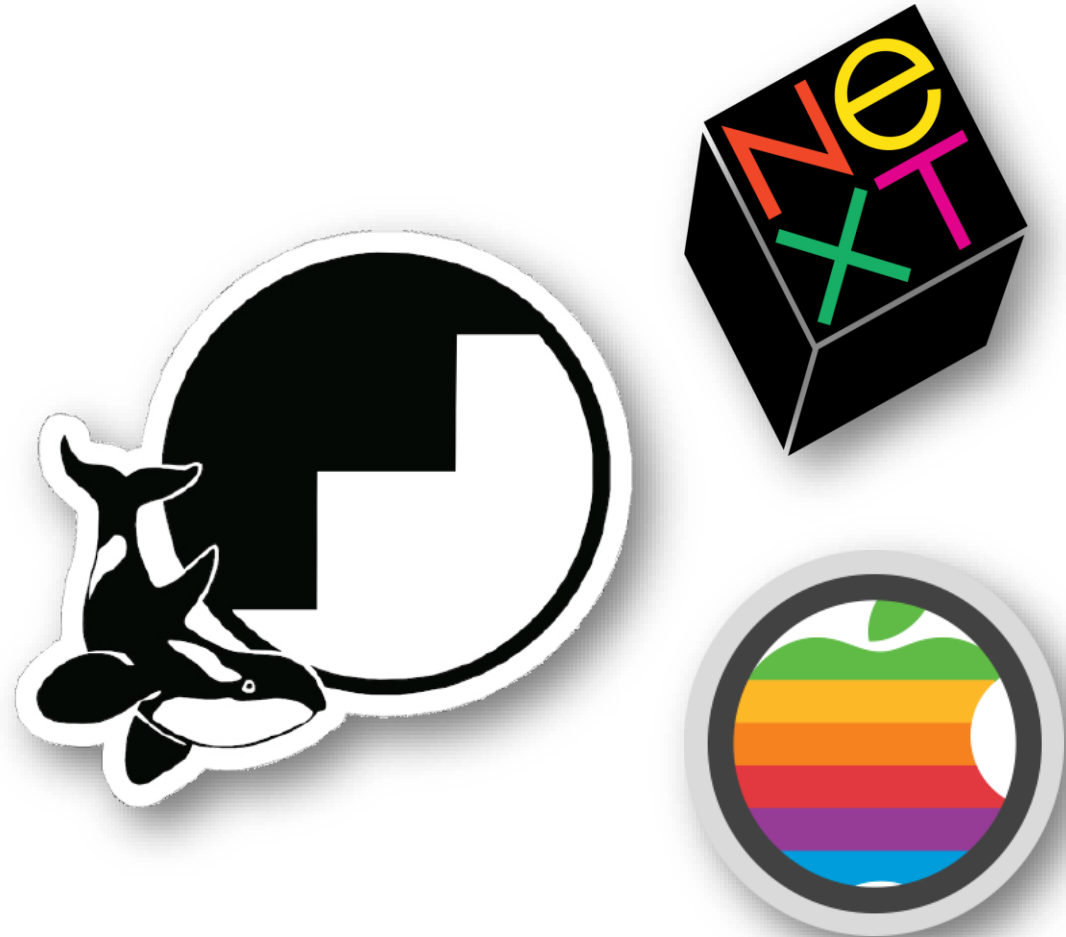
OVERVIEW OF GNUSTEP

- Open-source **framework** for developing **GUI applications**
- Designed to be a **cross-platform** implementation of **Apple's Cocoa** (OpenStep compliant)
- **Gorm**: interactive interface builder that helps developers build GUI applications using GNUstep's framework



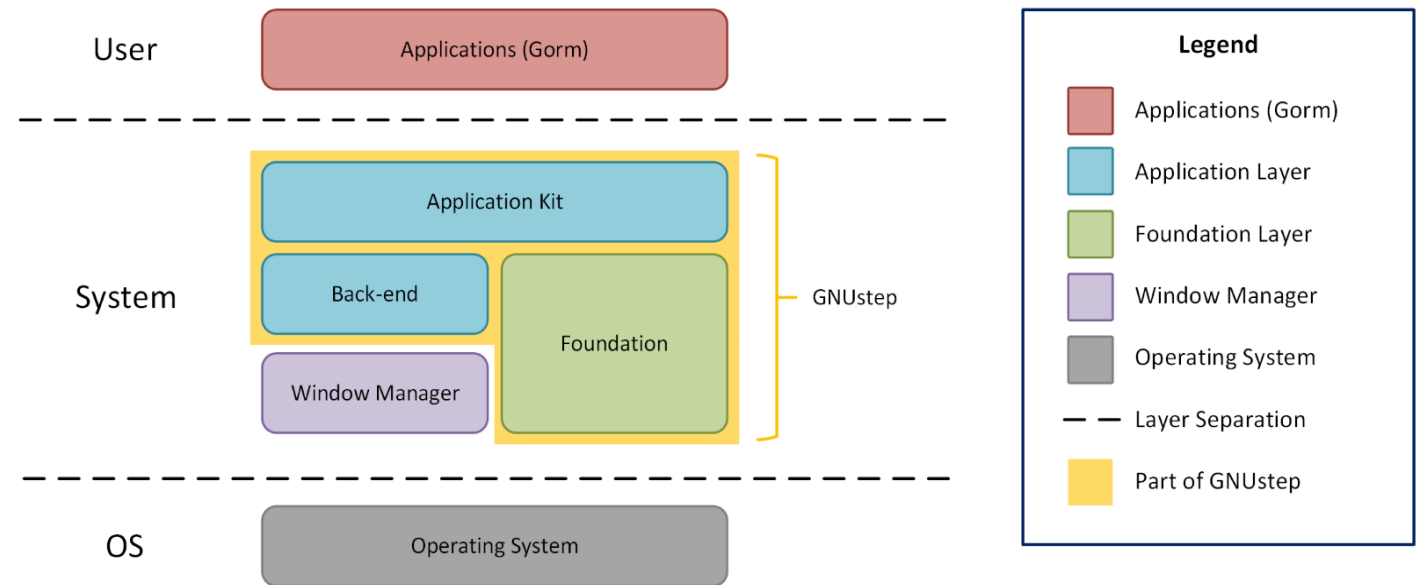
DERIVATION PROCESS

- **GNUstep Resources:** official website, wiki, system overview page
- **OpenStep Resources:** specification page
- **Apple Cocoa Resources:** developer websites, official documentation
- **Gorm Resources:** official application guide



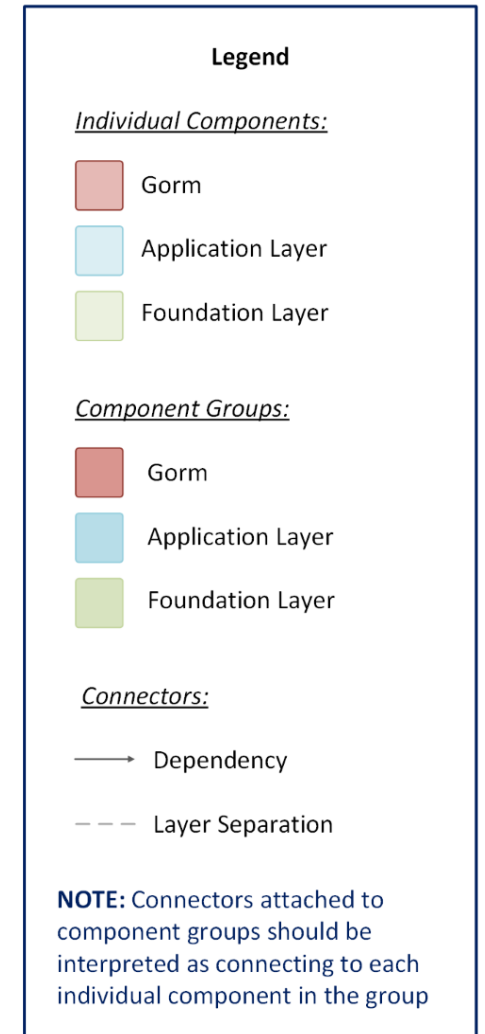
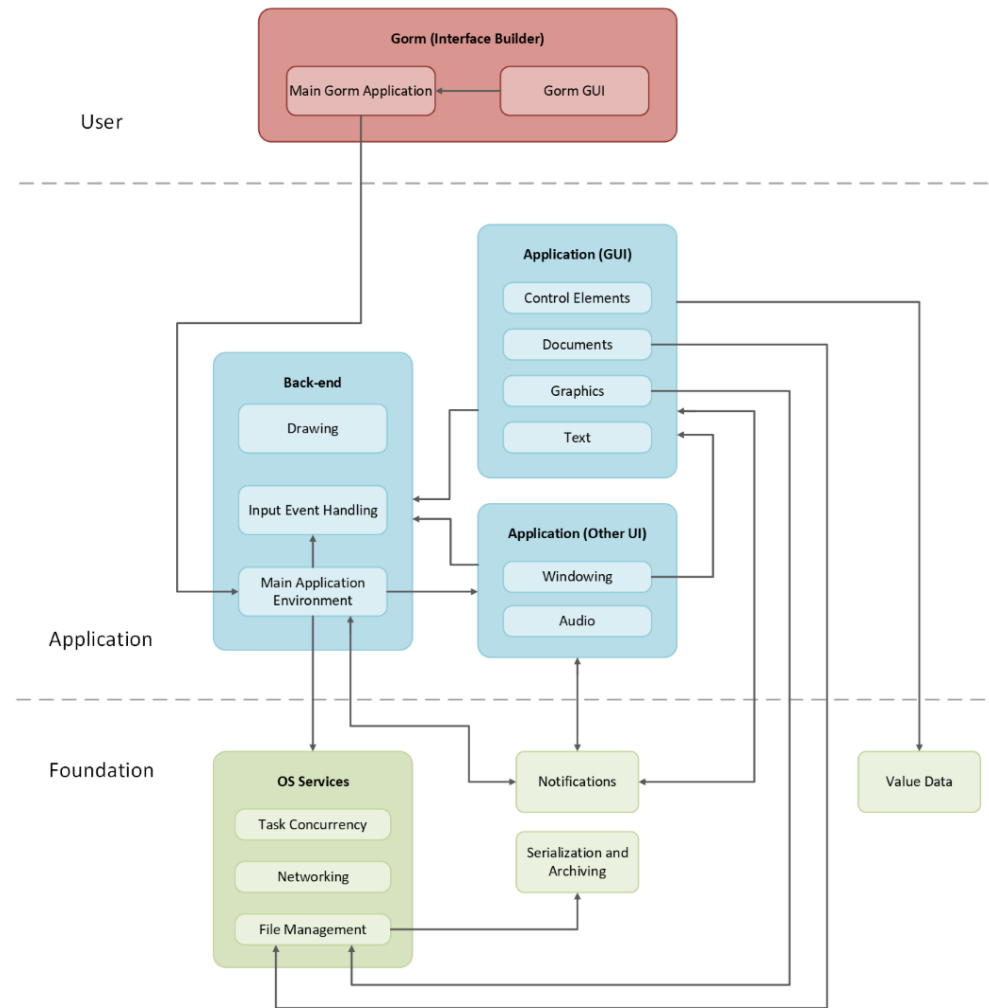
ARCHITECTURAL STYLES

- **Layered:** Foundation layer, Application layer, Gorm layer
- **Object-Oriented:** applications developed using GNUstep
- **Publish-Subscribe:** notification system for broadcasting events



COMPONENTS & DEPENDENCIES

- **Foundation:** Value Data, OS Services, Notifications, Serialization/ Archiving
- **Application:** Main Application Environment, Windowing, GUI Elements, Input Event Handling, Drawing, Audio
- **Gorm:** Gorm GUI, Main Gorm Application



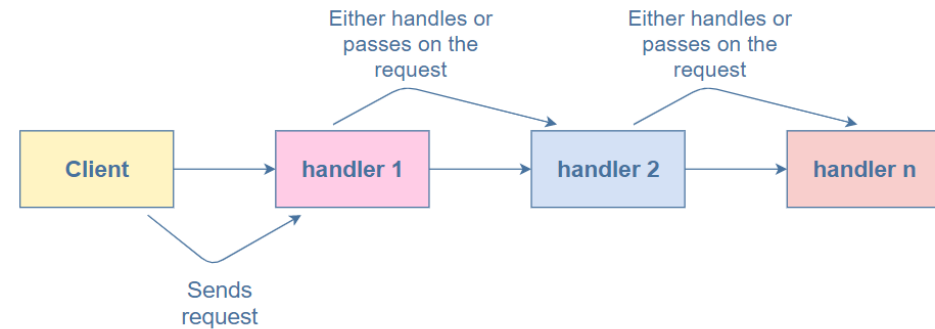
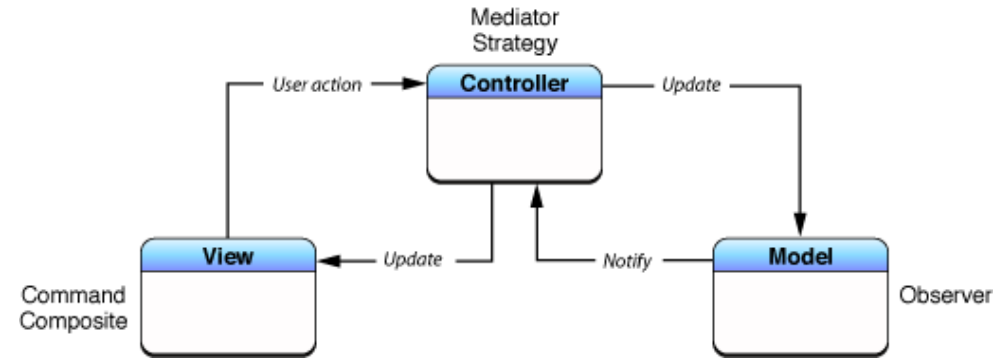
SYSTEM EVOLVABILITY

- Keeping up-to-date with **new features** **and updates** in Apple development tools
- Example: support for text-based elements in multilingual applications



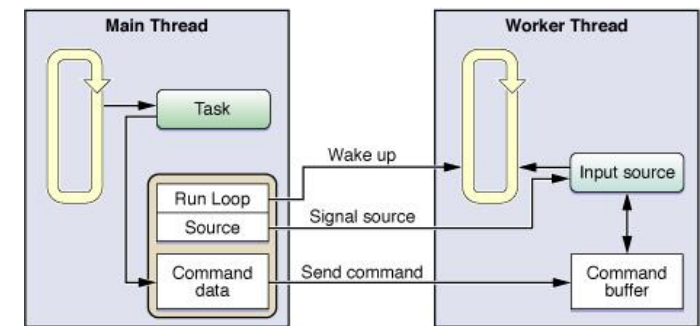
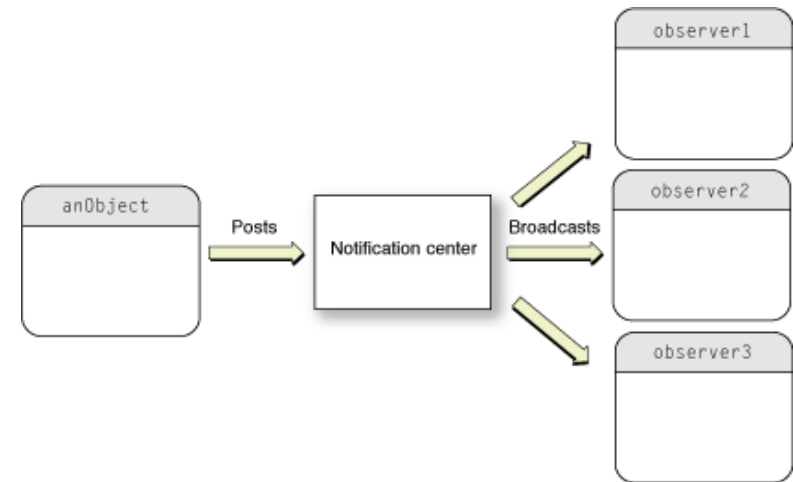
DESIGN PATTERNS

- **Model-View-Controller:** supported in applications developed using GNUstep
- **Chain of Responsibility:** implemented for handling user events



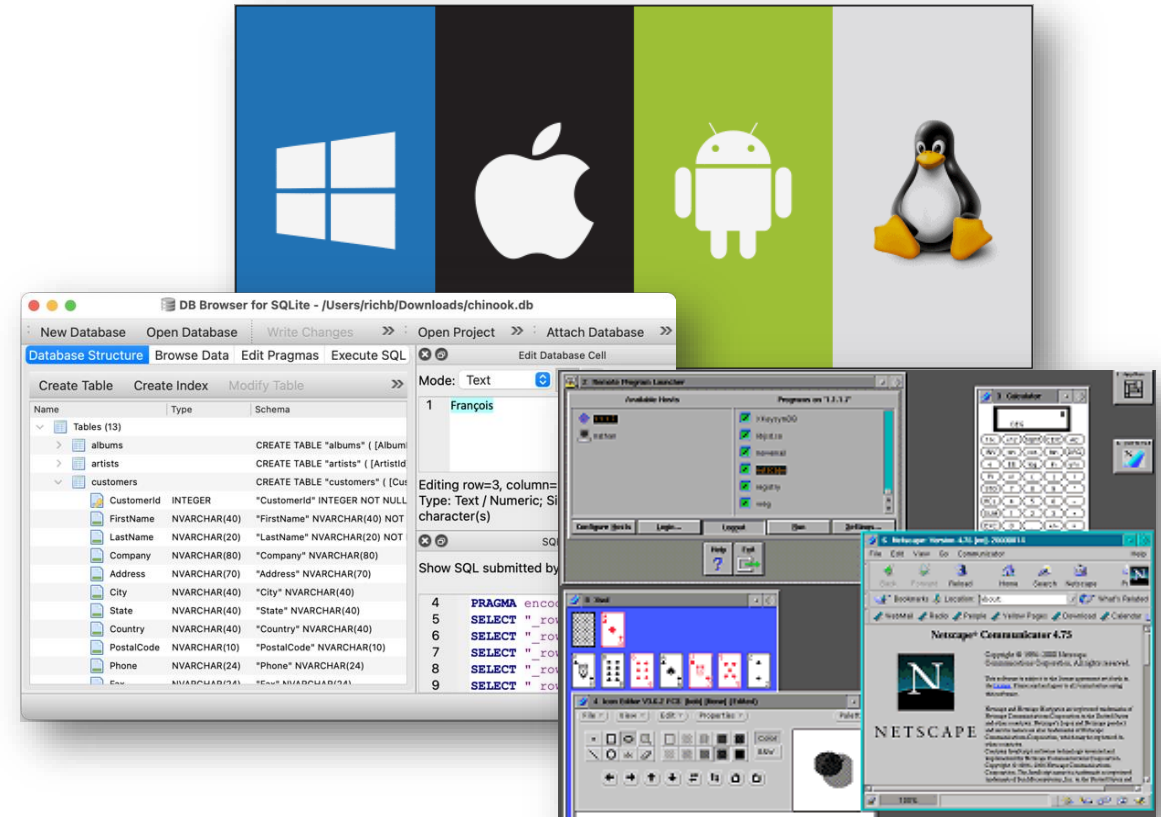
COMPONENT INTERACTIONS

- **Global Control Flow:** Chain of Responsibility, Notifications
- **Data Flow:** File Management
- **Concurrency:** Task Concurrency



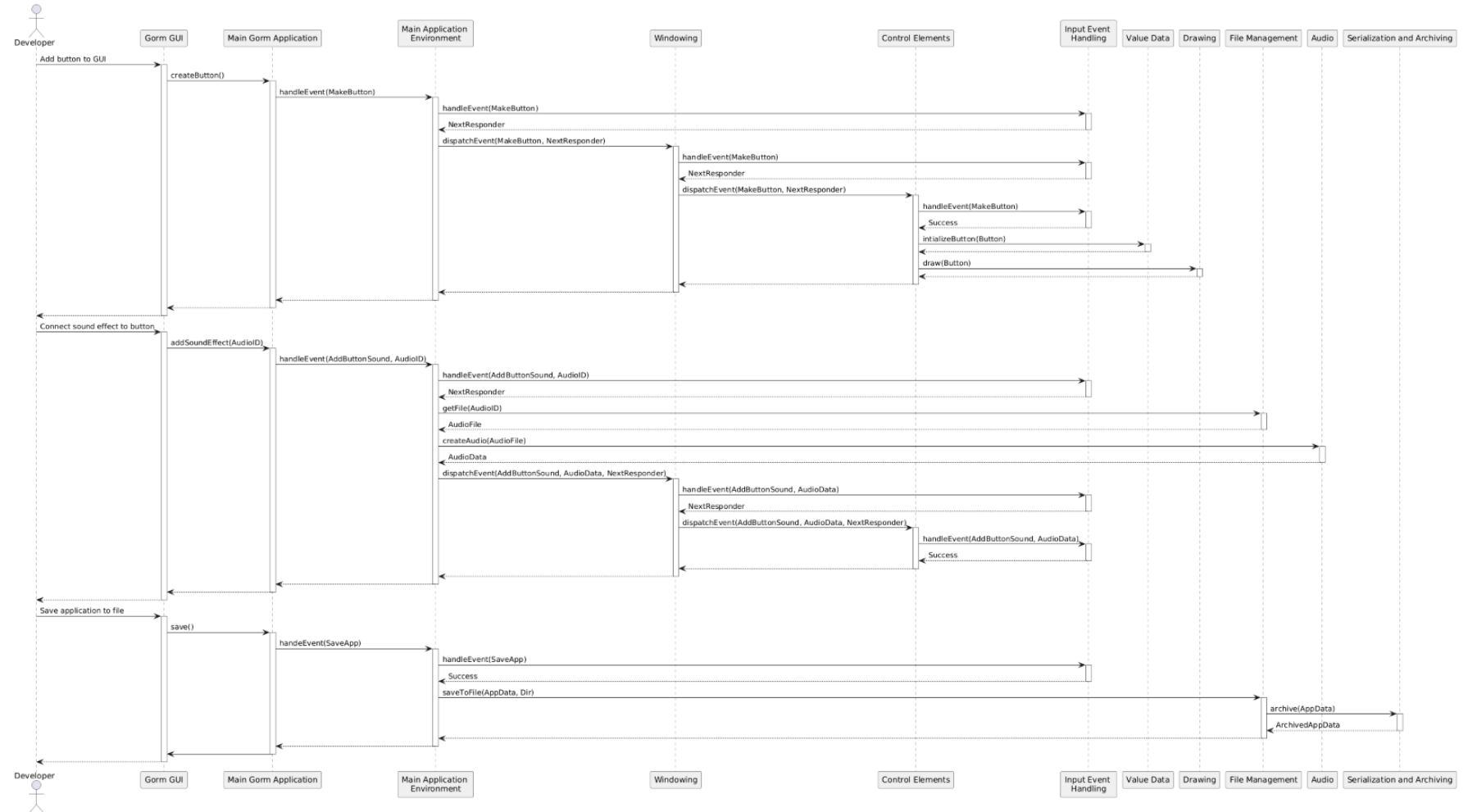
EXTERNAL INTERFACES

- **Operating System:** accessed through system calls and APIs implemented in GNUstep
- **Window Managers:** handle window placement, resizing, and decorations
- **Graphic Rendering:** manage graphical output to the machine
- **Packaging and Distribution:** aid in the deployment process of GNUstep applications



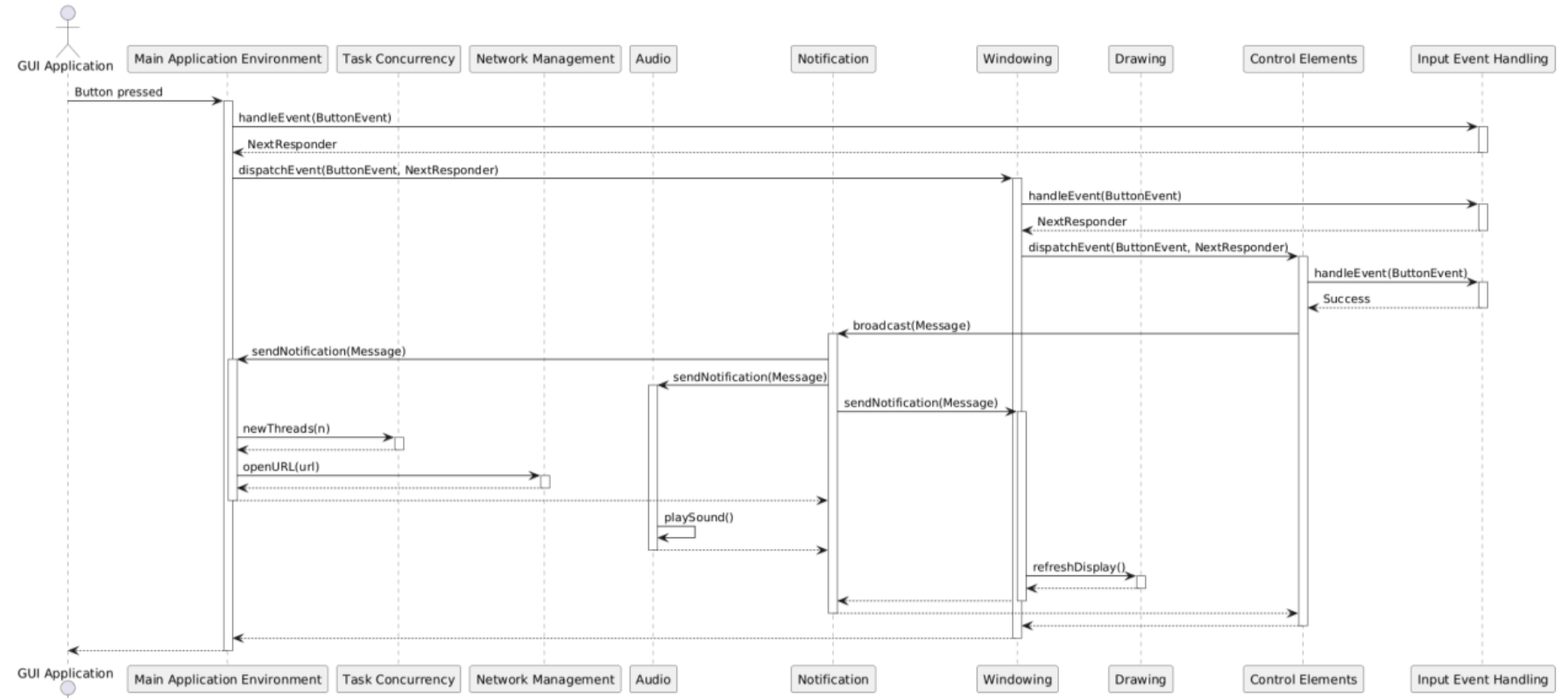
USE CASE 1:

Creating a button with
a clicking sound
effect in Gorm, and
saving the created
application



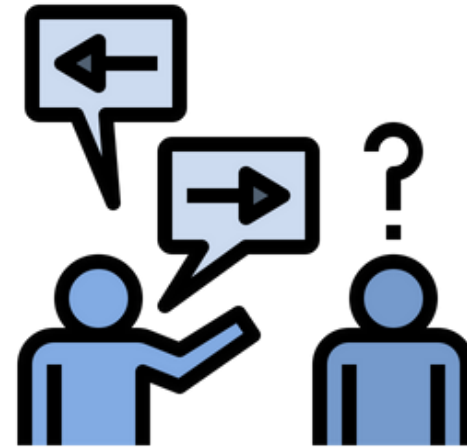
USE CASE 2:

Clicking a button in a
GNUstep application that
makes a sound effect
and opens a web link



LIMITATIONS

- **Sparse and incomplete** documentation for GNUstep
- Reliance on **OpenStep** and **Apple Cocoa** documentation for derivation
- Potential **inconsistencies** between the three sources



CONCLUSIONS & LESSONS LEARNED

- **Key features:** layered architecture, MVC design patterns, reliance on external interfaces
- **Bottleneck components:** Main Application Environment, Input Event Handling
- Must **evolve** alongside **Apple Cocoa** framework to remain relevant

