# Conceptual Architecture

url:https://youtu.be/Q86kXQMNe_E

**Group#8: 500 - Internal Server Error**

# Our Team

- **Group Leader:**

  Wanting Huang----(Intro & Abstract,  Lesson learned, Control and Flow of Data, External Interface)
- **Presenter:**

  Haoxi (Eleanor)Yang---- (Division of Responsibility)

  Qiu Xing (Nathan) Cai ----(Concurrency, )
- **Group Members**

  Yuda Hu ----(Evolution, Conclusion)

  Nicholas Wang ----(System Components & Interation)

  Amethyst Shen ----(Use Case & Sequence Diagram)

# Content

- Introduction of GNUstep
- Evolution
- Architectural  Style
- System Component & Interaction
- Use Case and Sequence Diagram
- Concurrency
- Division of Responsibility
- Lesson Learned

# Introduction

- Founded in 1993
- Open-source implementation of the OpenStep specification
- Follows NeXTSTEP and Cocoa API, offering a macOS-like development experience on other platforms
- Supports GUI and Server-side applications
- Architecture style ensures modularity and flexibility
- Very helpful for development

# Evolution

- Inception in 1993
- Evolved to adapt to modern development needs
- Designed as an open-source implementation
- Enhanced over time:
  -multithreading & asynchronous processing
  -modern data models in libs-corebase
  -development of Gorm
  -cross-platform improvements
- Evolved through community
  -GitHub
  -keep updates

# Architecture Style

## Layered Architecture

Libs-base(foundation layer)

libs-gui(application layer)

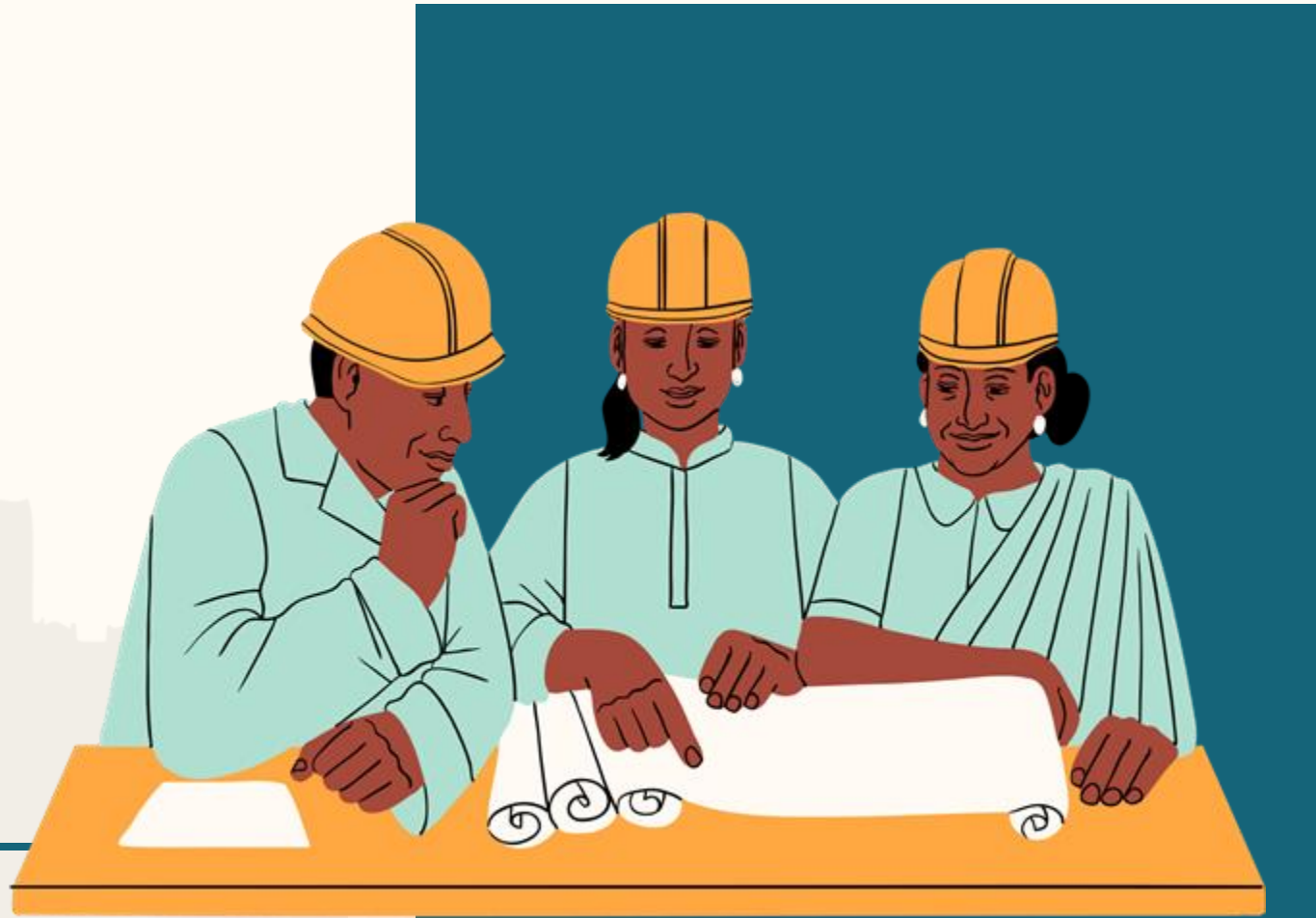
Libs-back(rendering layer)

## Object-Oriented

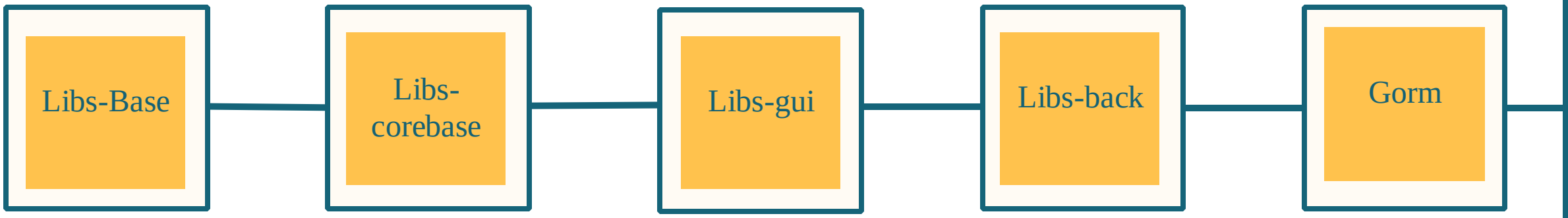
Encapsulation

Inheritance

Polymorphism



# System Component & Interaction




**Foundation Layer**

**System Utility Layer**

**GUI Layer**

**Rendering Layer**

**GUI Builder**

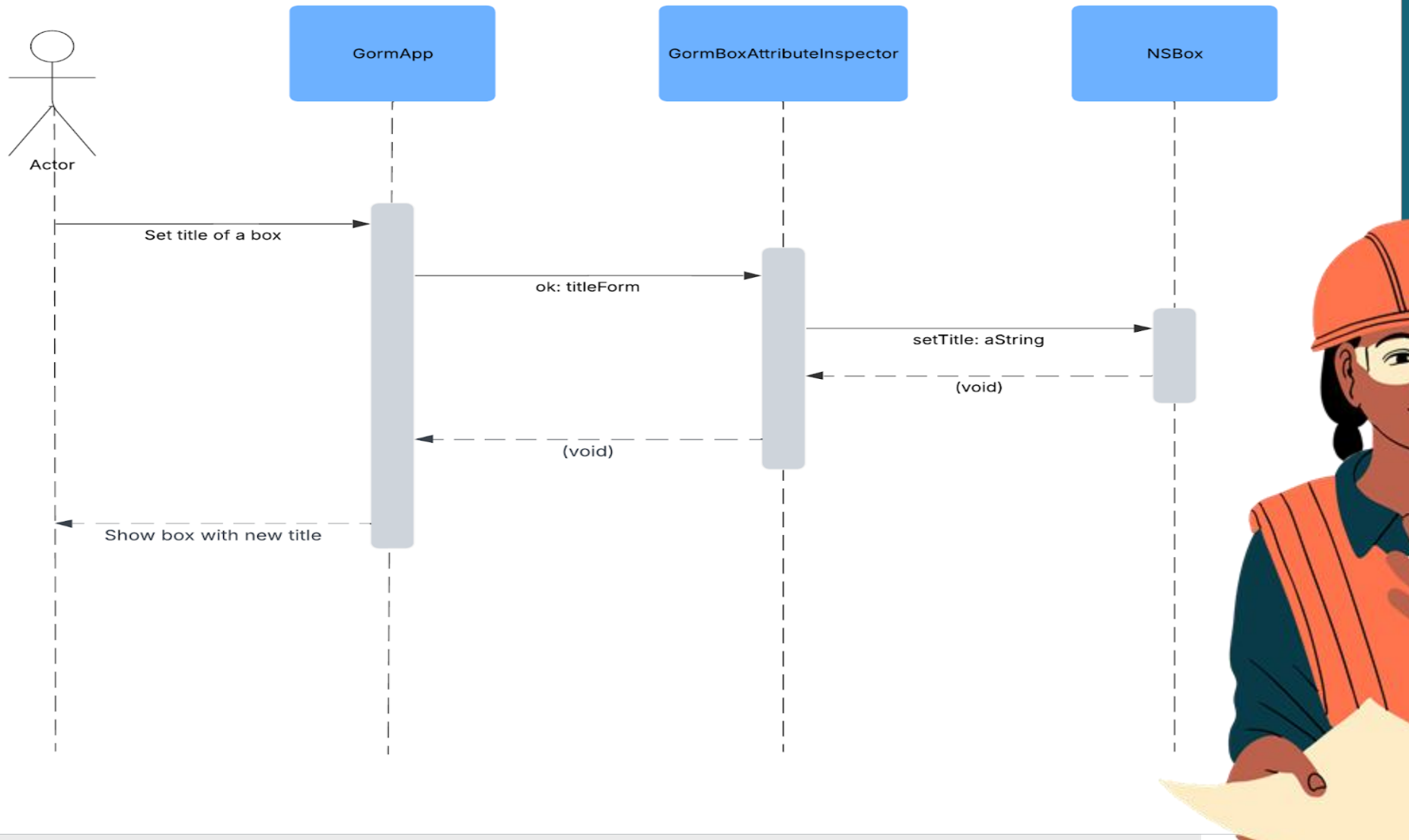
Libs-base & libs-core base -> system services

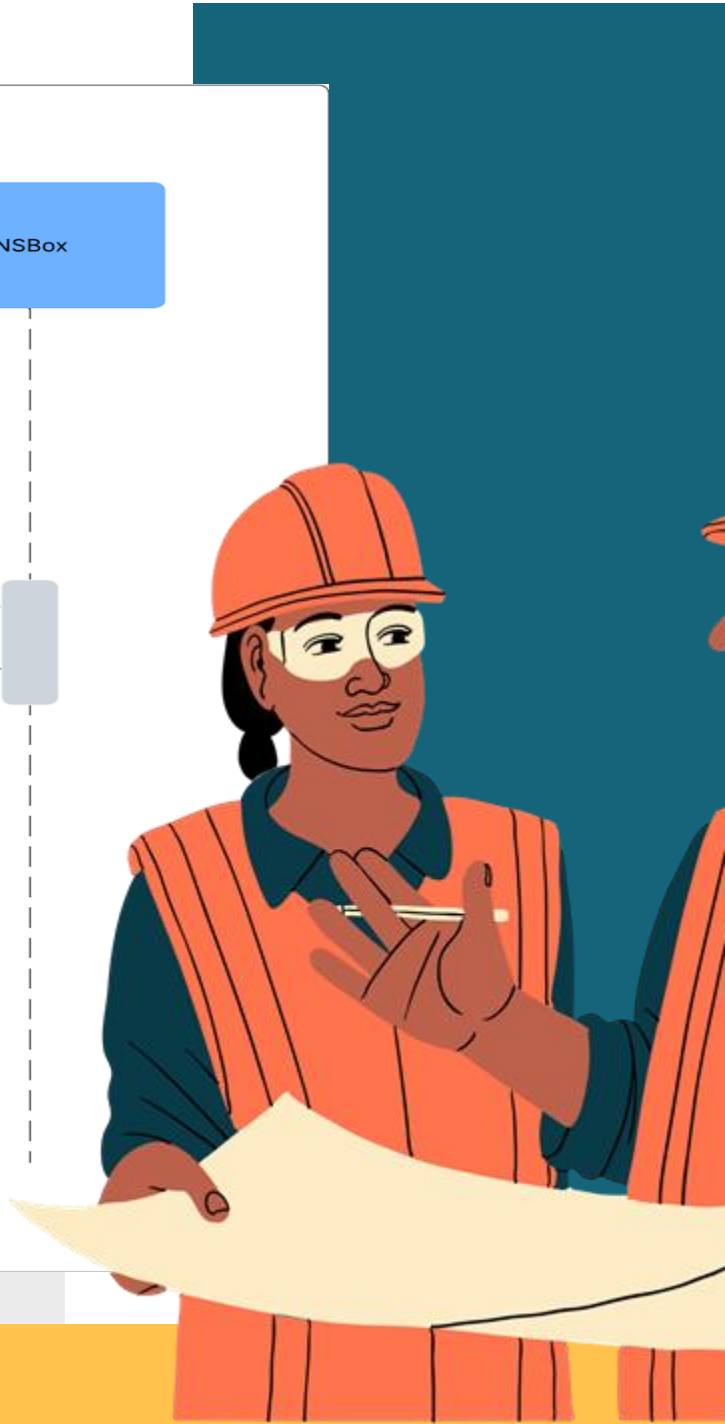
Libs-gui handles UI, delegates rendering to libs back

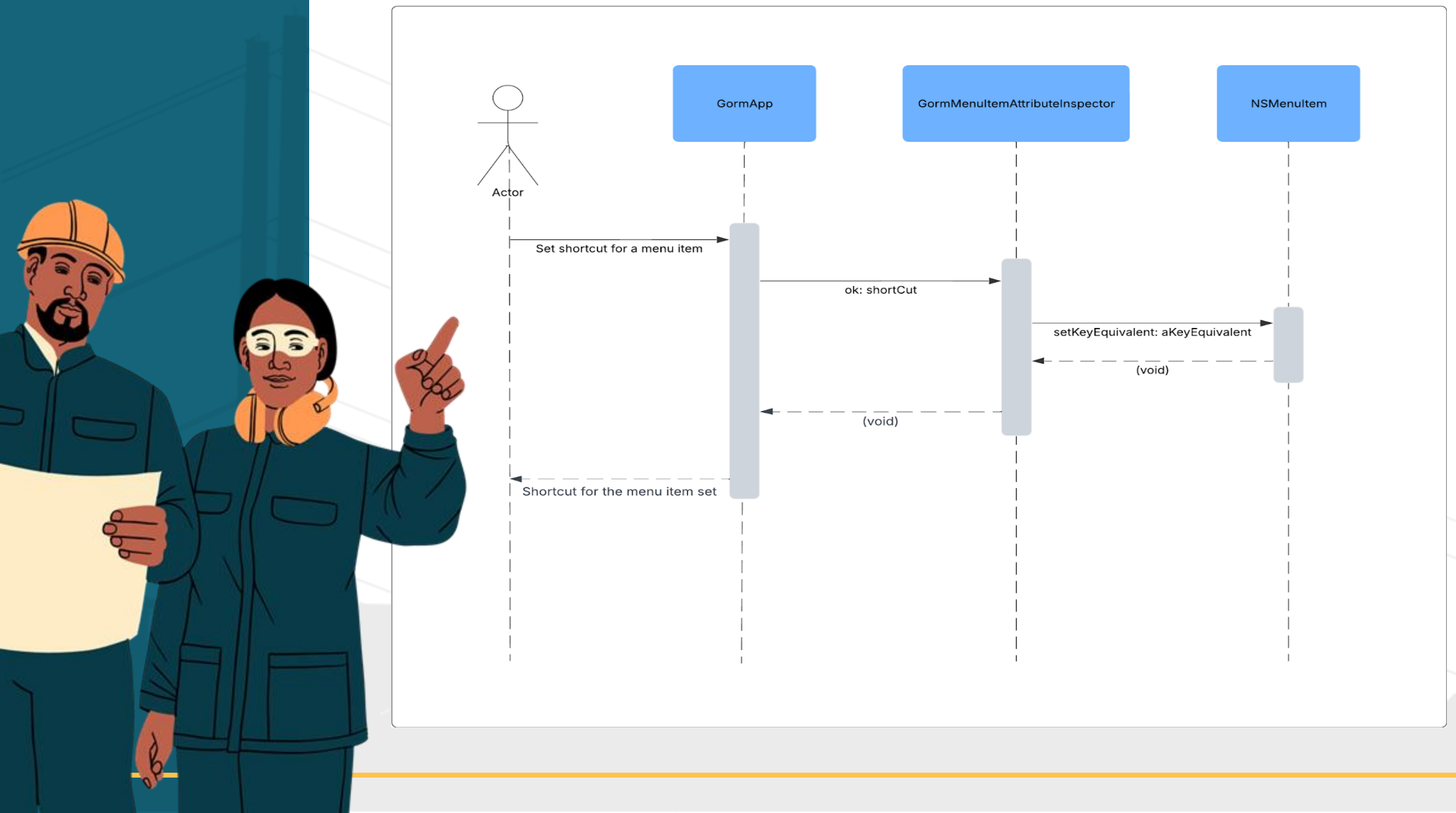
Libs-back interacts with OS-specific graphics

Gorm creates UI, processed dynamically by libs-gui

# Use Case and Sequence Diagram


GormApp
GormBoxAttributeInspector
NSBox
Actor
Set title of a box
ok: titleForm
setTitle: aString
(void)
(void)
Show box with new title




GormApp
GormMenuItemAttributeInspector
NSMenuItem
Actor
Set shortcut for a menu item
ok: shortCut
setKeyEquivalent: aKeyEquivalent
(void)
(void)
Shortcut for the menu item set


Gorm
Incremental Rendering
Drag-and-Drop
<<extend>>
<<include>>
Render Interface
Visual Object Configuration
<<include>>
<<include>>
<<include>>
Interactive Object Connection
Configuration
<<include>>
User
GNUstep
Real-Time Interface Testing
<<include>>
Load and Store
<<include>>
Project Persistence and Extensibility
<<extend>>
Serialize and Deserialize

# Concurrency

- GNUstep's libs-base library supports multithreading through the NSThread class
- But the code can run into problems with race conditions, where the threads manipulate the same data at the same time
- To prevent that we can use a Mutex lock provided by the NSLock and NSRecursiveLock

# Division of Responsibility: Documentation and Community Contributors

- Plays a critical role in the community
  - Writing and updating documentation
  - Answering questions from users

# Division of Responsibility: Core Library Developer

- Responsible for maintain and update the core libraries of GNUstep such as:
  - Libs-base
  - Libs-corebase
  - Libs-gui
- This includes
  - Low-level APIs in libs-corebase supporting GNUstep's higher level frameworks
  - Ensure compatibility with updating standards
  - Optimize libraries in performance and memory usage
  - Fix security vulnerabilities

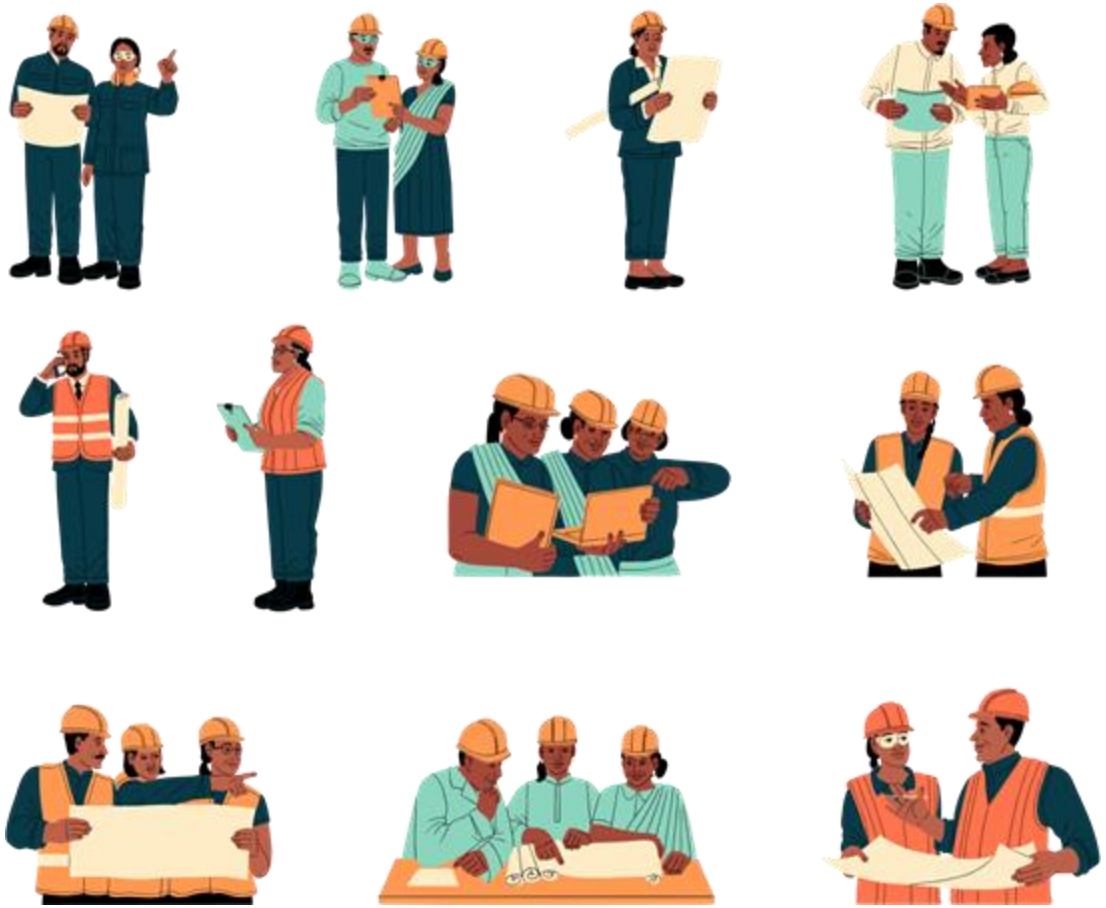
# Division of Responsibility:
# GUI and Tooling Devs

- Responsible for creating the GUI and development for GNUStep applications
- These developers develop and maintain the UI frameworks of GNUstep
- They use their UI design and Objective-C skills when developing for GNUstep ecosystem tools like Gorm, GNUstep ORM, etc.

# Conclusion

# Resource Page


-GNUstep source code repositories (libs-base, libs-gui, libs-back, libs-corebase, Gorm). https://github.com/gnustep
-GNUstep documentation on classes for concurrency. https://www.gnustep.org/resources/documentation/Developer/Base/Reference/NSThread.html
-Compatibility strategies for Cocoa integration. GNUstep Developer Documentation. https://www.gnustep.org/developers
-Gorm's transition to XLIF file formats. Gorm Manual. https://www.gnustep.org/resources/documentation/Gorm.pdf
-Strategies for legacy system support. GNUstep Wiki. https://mediawiki.gnustep.org
-Community-driven issue resolution processes. GNUstep Savannah. https://savannah.gnu.org/projects/gnustep
-Marcotte, L. (2002). *Programming under GNUstep: An Introduction*. *Linux Journal*, (108). Retrieved from https://dl.acm.org/doi/fullHtml/10.5555/640534.640540
-Wikipedia contributors. (2025, January 22). *GNUStep*. Wikipedia. https://en.wikipedia.org/wiki/GNUstep

# The End