Youtube link: [https://youtu.be/X2ECyu3zD6U](https://youtu.be/X2ECyu3zD6U)

# GNUstep Architecture

By:
Liam Beenken (Presenter #1, Slides, Control and Data Flow, Diagrams)
Cameron Jenkins (Presenter #2, Slides, Divisions of Responsibilities)
Evelyn Lee (Group Leader, Evolution, Abstract, Introduction, Sequence Diagram, Lesson Learned)
Christine Ye (System Functionality, Sequence Diagram, Data Dictionary)
Jihyeon Park (Concurrency, Conclusion)

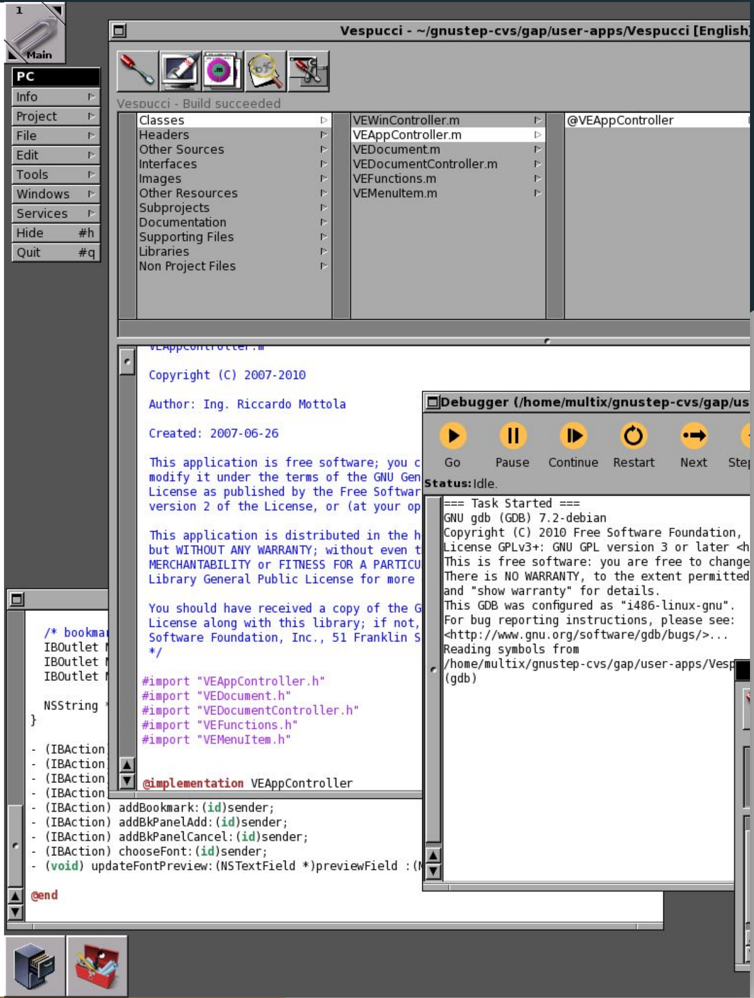
## Component Overview

## Components:

- ProjectCenter
- Gorm (Graphical Object Relationship Modeller)
- GNUstep-CoreBase
- GNUstep-Base
- GNUstep GUI Library
- GNUstep GUI Backend

## ProjectCenter

- Integrated IDE
- Manage Projects
- Makefile Generation
- Write and test applications



Vespucci - ~/gnustep-cvs/gap/user-apps/Vespucci [English]

Vespucci - Build succeeded

| | | |
|---|---|---|
| Classes | VEWinController.m | @VEAppController |
| Headers | VEAppController.m | |
| Other Sources | VEDocument.m | |
| Interfaces | VEDocumentController.m | |
| Images | VEFunctions.m | |
| Other Resources | VEMenuItem.m | |
| Subprojects | | |
| Documentation | | |
| Supporting Files | | |
| Libraries | | |
| Non Project Files | | |

```
   Copyright (C) 2007-2010

   Author: Ing. Riccardo Mottola

   Created: 2007-06-26

   This application is free software; you c
   modify it under the terms of the GNU Gen
   License as published by the Free Softwar
   version 2 of the License, or (at your op

   This application is distributed in the h
   but WITHOUT ANY WARRANTY; without even t
   MERCHANTABILITY or FITNESS FOR A PARTICU
   Library General Public License for more

   You should have received a copy of the G
   License along with this library; if not,
   Software Foundation, Inc., 51 Franklin S
   */

#import "VEAppController.h"
#import "VEDocument.h"
#import "VEDocumentController.h"
#import "VEFunctions.h"
#import "VEMenuItem.h"


@implementation VEAppController
```

Debugger (/home/multix/gnustep-cvs/gap/us

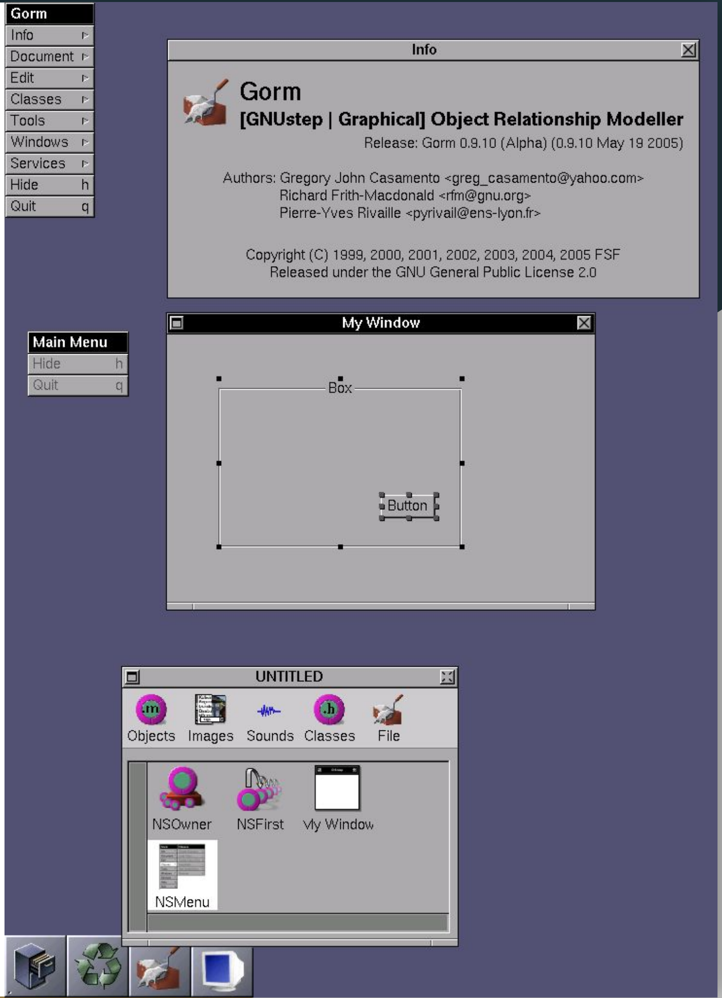
Status: Idle.

```
=== Task Started ===
GNU gdb (GDB) 7.2-debian
Copyright (C) 2010 Free Software Foundation,
License GPLv3+: GNU GPL version 3 or later <h
This is free software: you are free to change
There is NO WARRANTY, to the extent permitted
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from
/home/multix/gnustep-cvs/gap/user-apps/Vesp
(gdb)
```

```
  /* bookma
  IBOutlet
  IBOutlet
  IBOutlet

  NSString
}

- (IBAction)
- (IBAction)
- (IBAction)
- (IBAction)
- (IBAction) addBookmark:(id)sender;
- (IBAction) addBkPanelAdd:(id)sender;
- (IBAction) addBkPanelCancel:(id)sender;
- (IBAction) chooseFont:(id)sender;
- (void) updateFontPreview:(NSTextField *)previewField :(N

@end
```

## Gorm

- Graphical tool for UI design and implementation
- Simplifies interface creation
- Reduces code
- Creation of .gorm files, used to load your GUIs into your application
- Mimic’s Apple’s InterfaceBuilder



Gorm
[GNUstep | Graphical] Object Relationship Modeller
Release: Gorm 0.9.10 (Alpha) (0.9.10 May 19 2005)

Authors: Gregory John Casamento <greg_casamento@yahoo.com>
Richard Frith-Macdonald <rfm@gnu.org>
Pierre-Yves Rivaille <pyrivail@ens-lyon.fr>

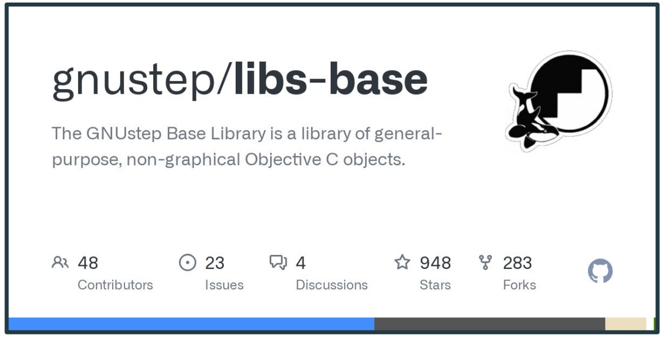
Copyright (C) 1999, 2000, 2001, 2002, 2003, 2004, 2005 FSF
Released under the GNU General Public License 2.0

## GNUstep Core-Base

- Designed for compatibility with Apple’s CoreFoundation
- Suitable for lower level projects, and performance optimization
- Low level wrapper for C functionality
- Abstracts C concepts, providing useful higher level structures like Strings, Arrays, and Event Loops
- Non-Graphical

## GNUstep Base

- Similarly abstracts lower level concepts
- Objective-C general purpose library
- Slightly slower than CoreBase, but much more programmer-friendly
- Provides useful Objective-C objects and APIs
- Non-Graphical
- Suitable in most application development cases
- Designed for compatibility with Apple’s Foundation module


gnustep/**libs-base**

The GNUstep Base Library is a library of general-purpose, non-graphical Objective C objects.

48 Contributors · 23 Issues · 4 Discussions · 948 Stars · 283 Forks

## GNUstep GUI Library

- Provides GUI components for development
  - Buttons, text fields, etc.
- Provides API for handling actions on the components
- Written in Objective-C

gnustep/libs-gui

The GNUstep gui library is a library of graphical user interface classes written completely in the Objective-C language; the classes...

39 Contributors · 41 Issues · 1 Discussion · 281 Stars · 103 Forks

## GNUstep GUI Backend

- Backend for the GUI library
- Handles component rendering
- Accounts for differences across machines and platforms
- Communicates with system's window manager to properly display applications
- Maintains a consistent look on every system

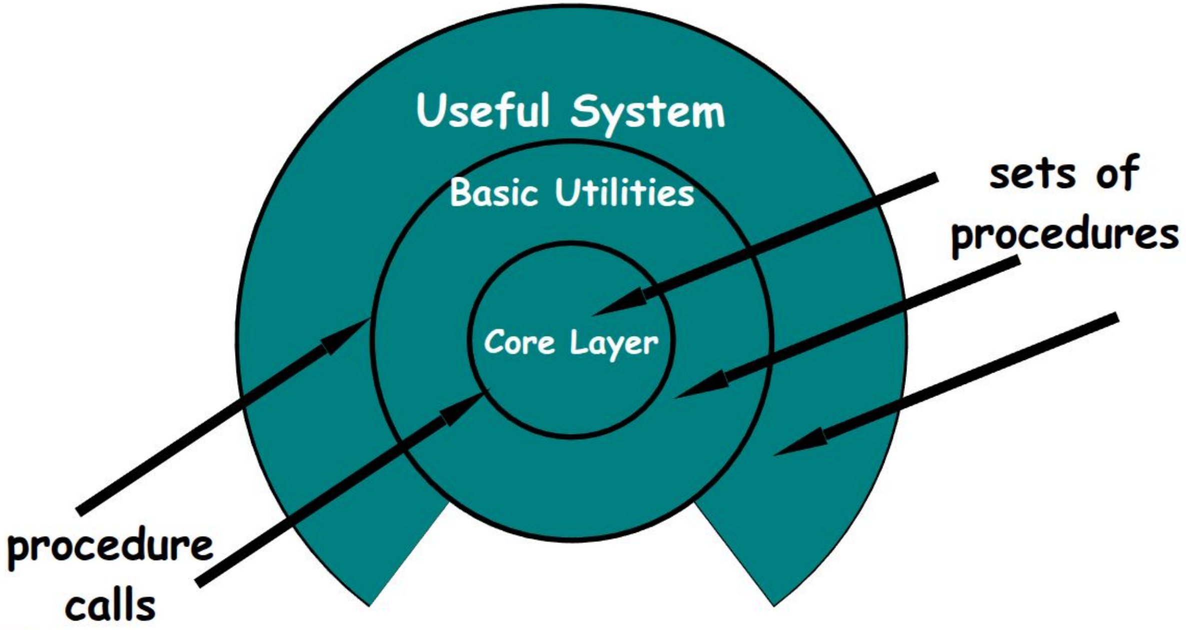
gnustep/**libs-back**

The GNUstep gui library is a library of graphical user interface classes written completely in the Objective-C language; the classes...

28 Contributors · 6 Issues · 50 Stars · 34 Forks

## Top Level Architecture

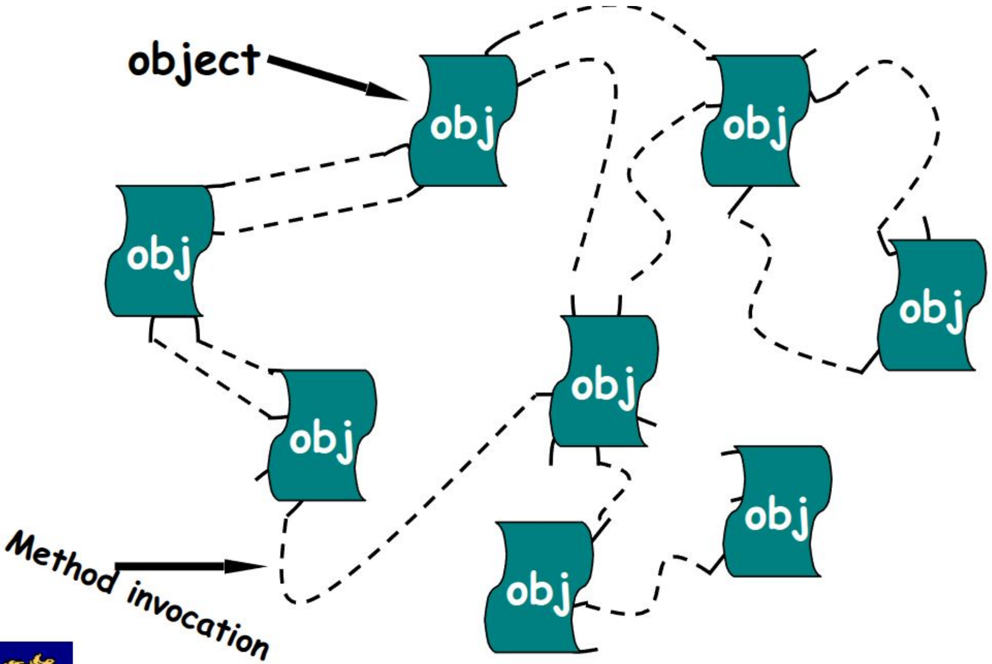
## Layered Style



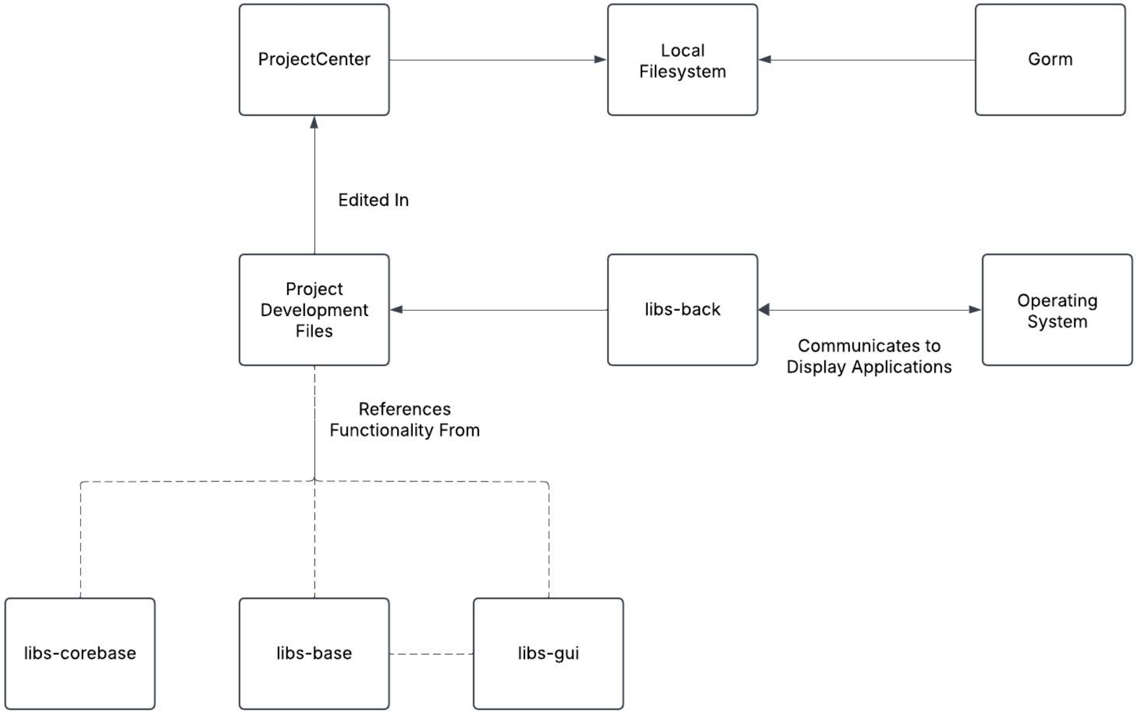
## Layering

- A large portion of GNUstep is built upon itself
- Base and corebase provide the core of the application - similar to C/Objective-C standard library
- Most of GNUstep will rely on one of these two libraries
- The GUI library is built on top of base
- We then see built applications designed and run on top of the GUI libraries
- ProjectCenter and GORM are examples of useful systems built using these lower layers
- We add further layering when we design custom applications using ProjectCenter

## Object Oriented Style

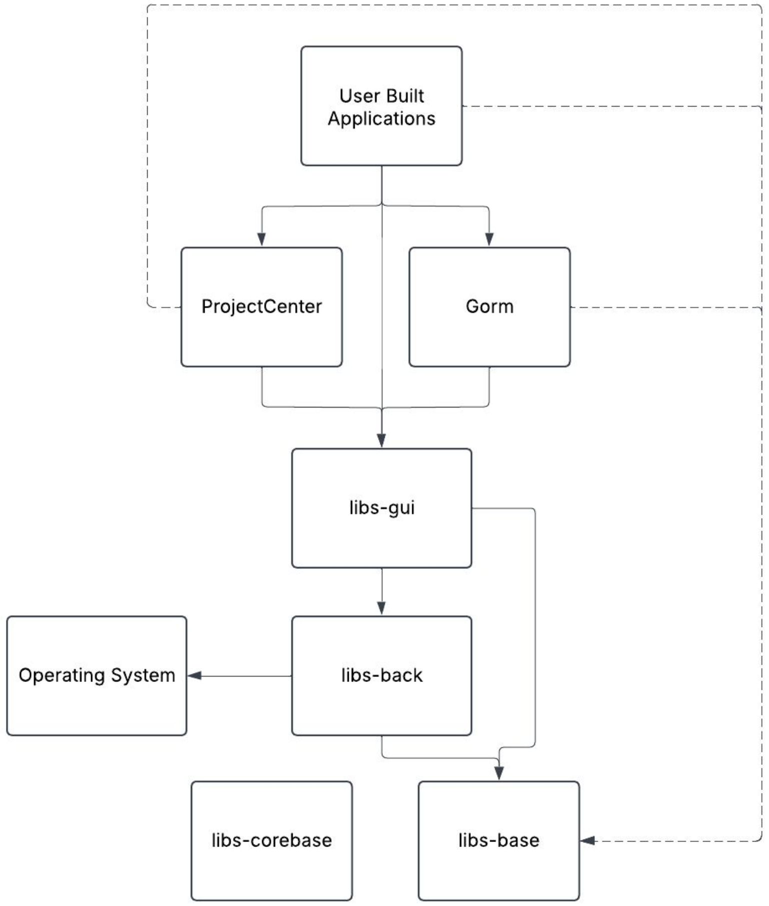


# Development Workflow



## Conceptual Architecture

- Clear layering of components
- Each level uses only functionality from the levels below
- In some cases, higher components access base for efficiency and saved development time
- Each component exposes an API for the others to access
- Most components encapsulate their data in objects



## Concurrency and Team Implications

Effects of Concurrency on GNUstep:

- Multithreading Challenges, GUIs require smooth responses, so threading must carefully be managed so it can avoid race conditions.
- GNUstep uses an event loop to handle user inputs and system events efficiently.
- Component interactions, Backend communication, event handling and GUI rendering all require proper synchronization to function.

Team implications:

- Teams working on different components must coordinate their threading/ multithreading models.
- API boundaries can prevent concurrency bugs across development teams.

## Lessons Learned

- Concurrency must be planned early. It is vital for synchronization mechanisms to be designed from the beginning.
- Ensuring GNUstep is compatible with macOS shows  tradeoffs with performance and also flexibility.
- Documenting properly is critical. Properly documented open source projects help contributors add their work more smoothly.
- Separating the CoreBase, Base, GUI and Backend prevents unnecessary dependencies (Modularity).

## Derivation Process and Alternatives

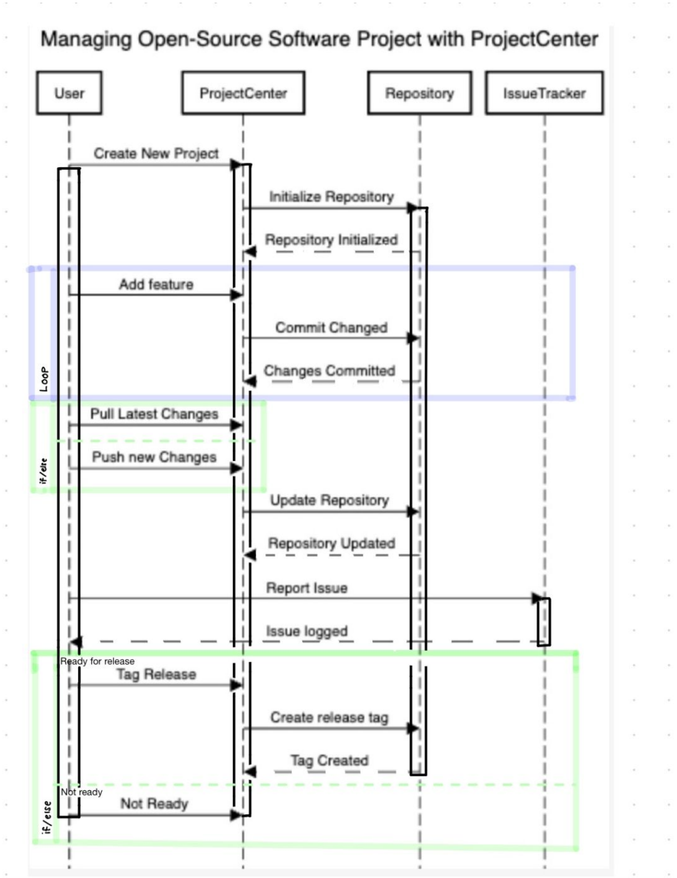
Derivation Process:

- GNUstep is based on the NeXTSTEP and OpenStep operating systems.
- They use a layered approach with Corebase and Base providing Objective-C functionality while the GUI builds over top. Dynamic linking in Objective-C allows the base/core-base functions to be called at runtime.

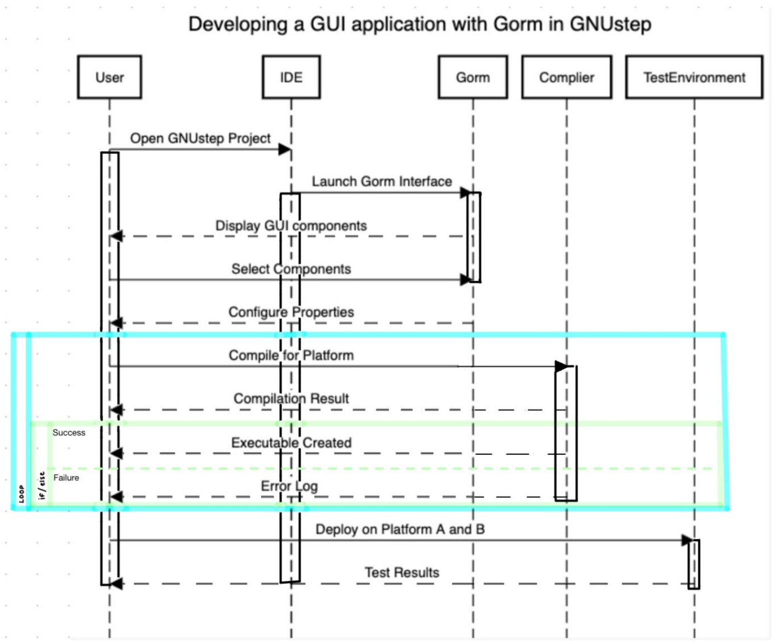
Alternatives Considered in the derivation process:

- Integrating with GTK or Qt, this could have provided a wider range of widgets and likely could have resulted in more developers using GNUstep earlier, although this approach could have led to compromises with their design and potential performance inconsistencies.

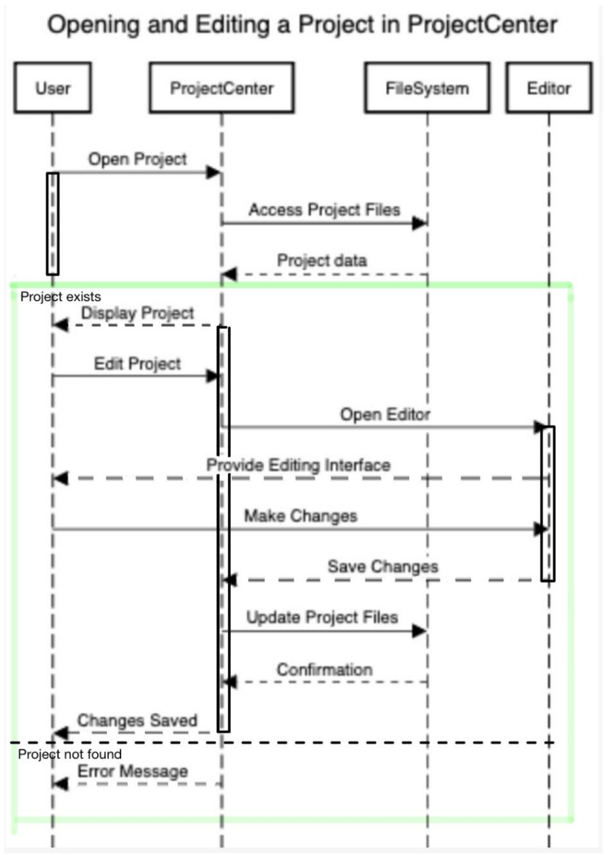
# Sequence Diagram (Managing Open-Source Software Project with ProjectCenter)



## Sequence Diagram (Developing a GUI application with Gorn in GNUstep)



## Sequence Diagram (Opening and Editing a Project in ProjectCenter)



## Conclusion

- GNUstep is an open-source, cross-platform framework based on OpenStep and Cocoa.
- It provides essential libraries (Base, GUI, CoreBase) and development tools (ProjectCenter, Gorm).
- It’s evolved to improve  Cocoa compatibility, Objective-C support, and UI rendering.
- Enables dynamic linking, cross-platform GUI development, and networking support.
- Due to it being open-source it can continue to adapt to the communities needs with more contributions.