# Soar General Input/Output Interface (SGIO) An Overview

By Brad Jones

# Motivation

- Embedding a must
- Library for faster development
- Needed to make SocketIO easier to use and understand

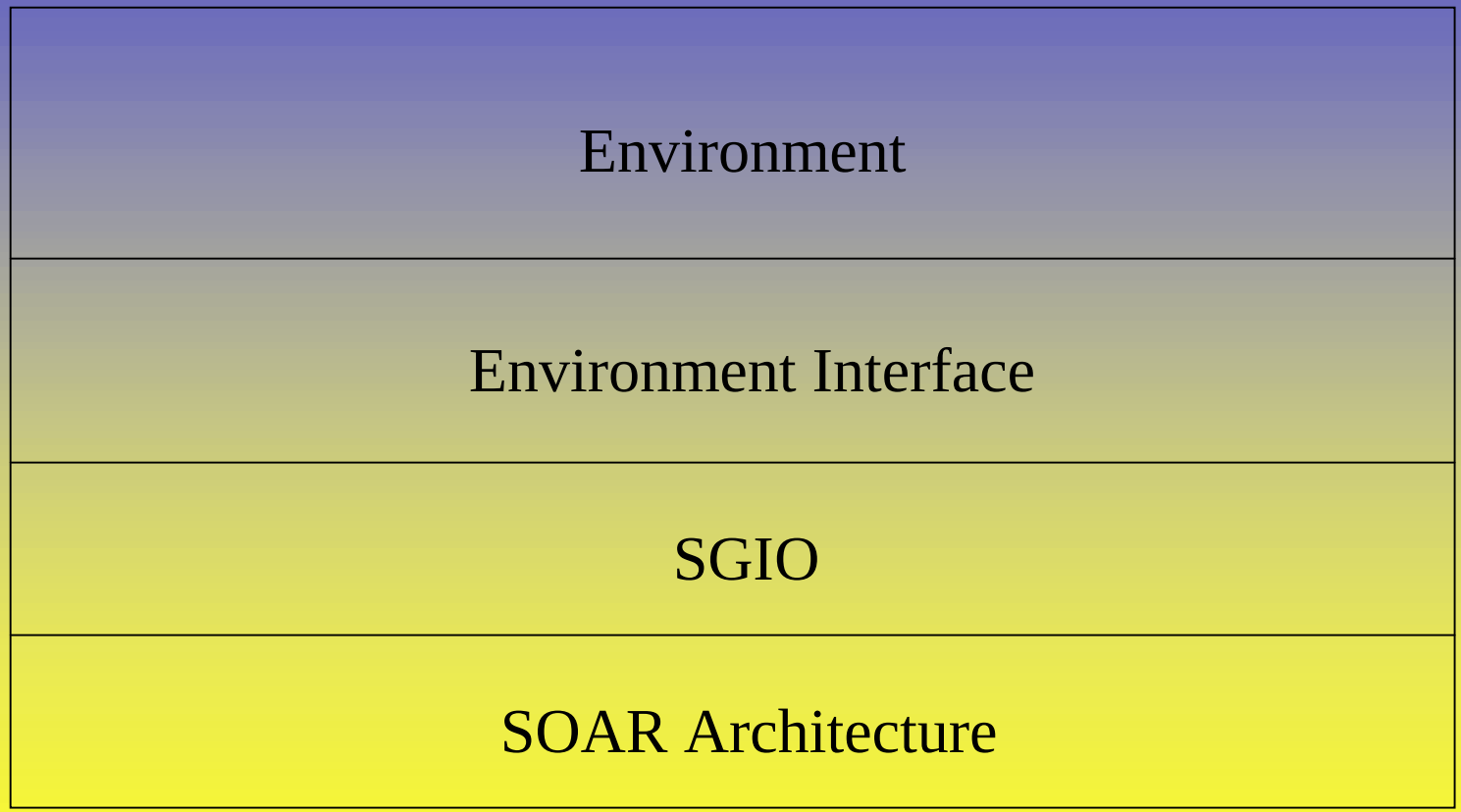
# Design Goals

- Incorporate SocketIO
- Allow the user to embed the SOAR Architecture within the environment
- This change in communication with the SOAR Architecture should require minimal changes to the code
- The Quake2 must play well using this library

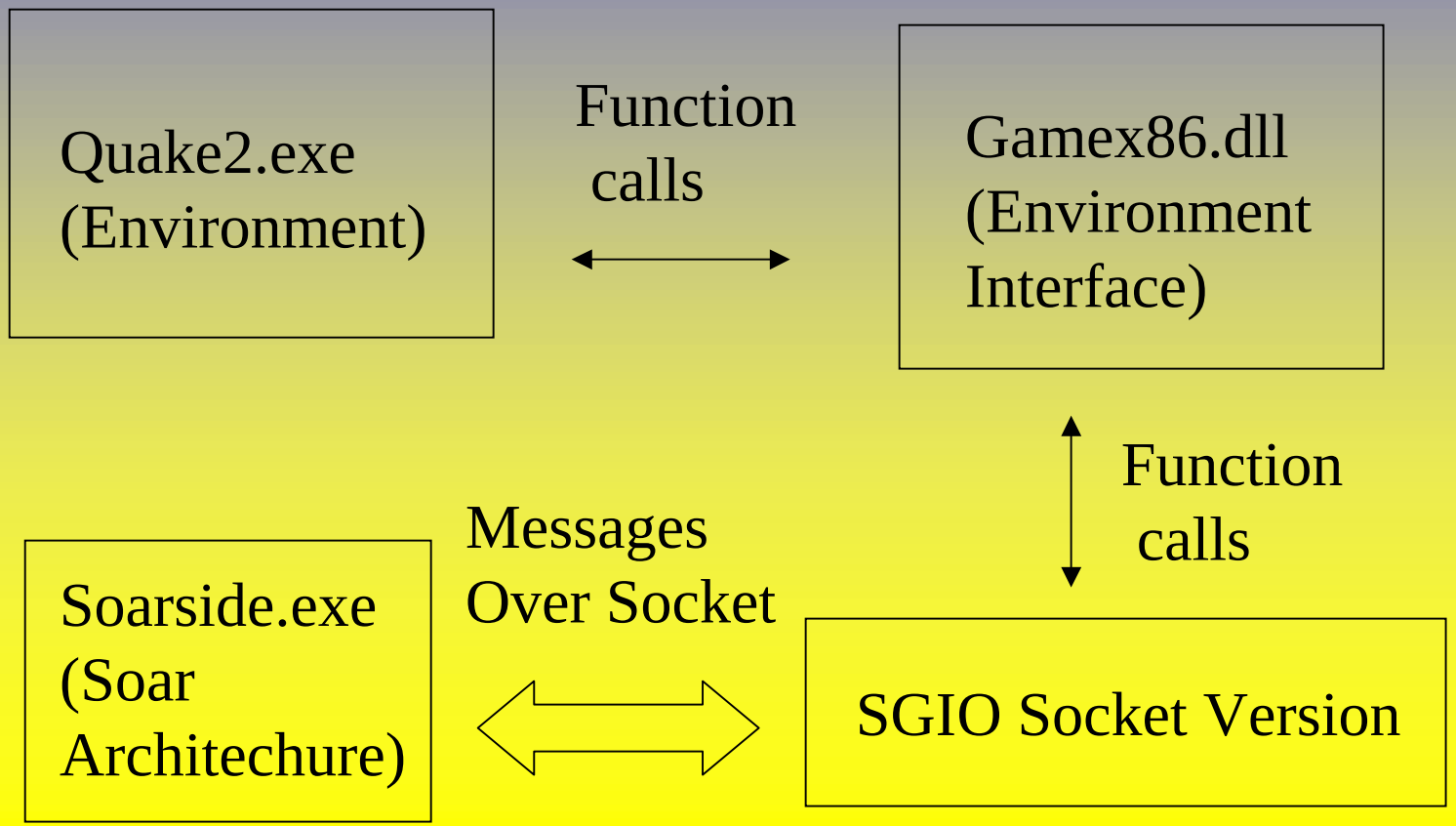
# SGIO Instantiated

- SGIO is a C++ interface library to handle SOAR Architecture input/output
- Mediator between the environment and the SOAR Architecture
- Run-time choosing of type of communication

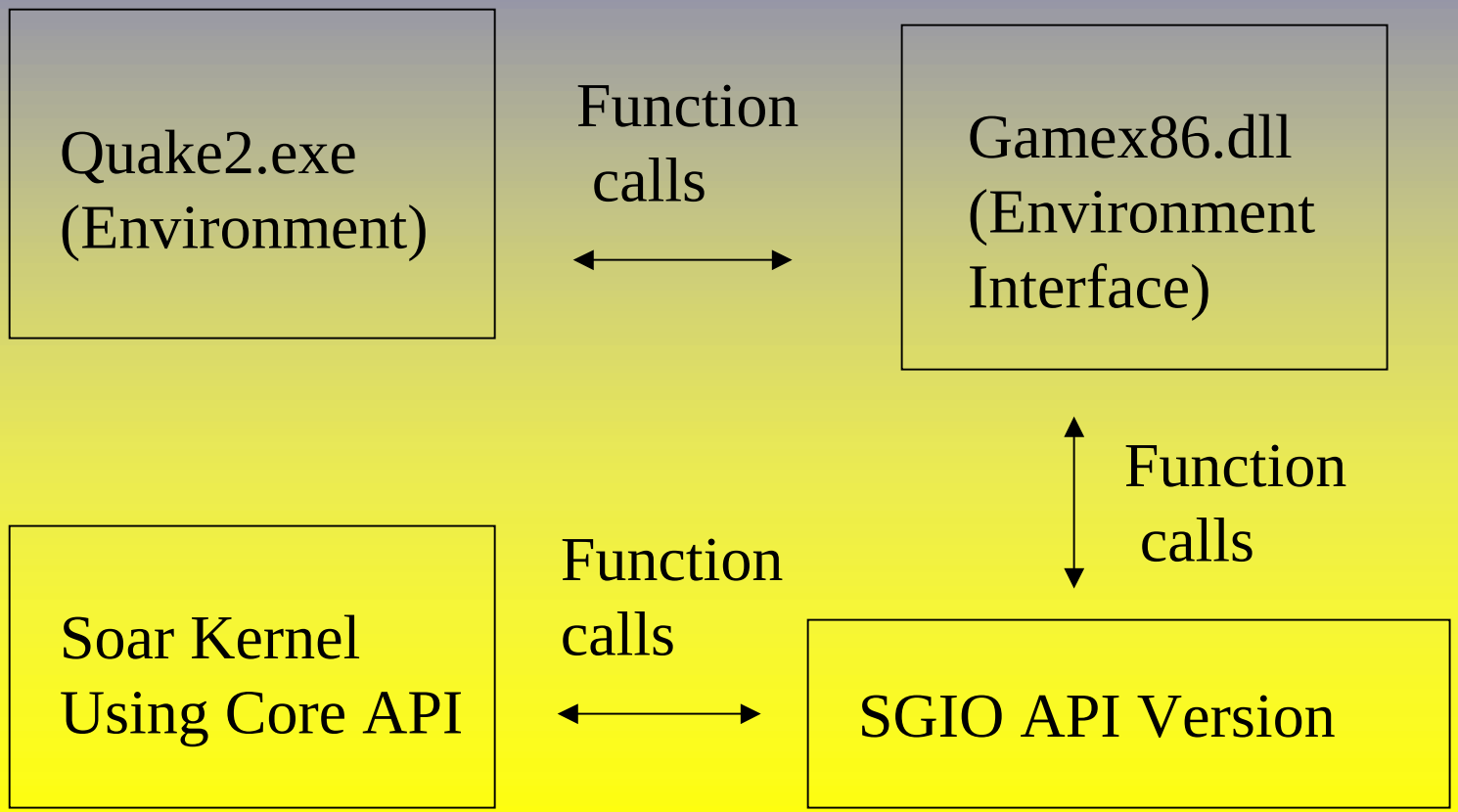
# Functional Boundaries



# Quake2
# Socket Communication



# Quake2
# Functional Communication



# Changes to Code Needed to Use Socket Version Vs. API Version

- Socket Version

  soar = new SIOSoar(
  “141.213.12.133”,
  7070);

- API Version

  soar = new APISoar();

# Main Components of SGIO

- Soar class
  - Establishes communication with SOAR Architecture
- Agent class
  - Proxy for SOAR Architecture Agent
- WorkingMemory class
  - Layer on top of Agent, just deals with the Agent interface

# Soar Class (SGIO)

- Sets up some type of communication with SOAR Architecture
  - APISoar uses the core api
  - SIOSoar uses a socket interface
  - LogSoar logs calls that you make into the interface for debugging

# Example Functions: Soar Class

- RunTilOutput
- CreateAgent
- DestroyAgent

# Agent Class (SGIO)

- Acts as a proxy
- Keeps WME changes until the agent actually needs them
- Keeps a queue of commands that the Agent has issued until the environment can process them

# Example Functions: Agent Class

- AddWME
- RemoveWME
- CommitWMEChanges
- LoadProductions
- Commands
- GetCommand

# WorkingMemory Class (SGIO)

- Layers on top of the Agent Class
- Takes care of a lot of the bookkeeping associated with Working Memory elements

| | Agent Class |
|---|---|
| WorkingMemory Class Uses | AddWME<br>RemoveWME<br>CommitWMEChanges |
| | LoadProductions<br>Commands<br>GetCommand |

# Example Functions: Working Memory Class

- Update
- CreateFloatWME
- CreateIntWME
- CreateIdWME
- CreateStringWME
- Commit

# Future Work

- Regression Validation of SGIO
- Performance Measurements
- Debugging Aids
- Better Documentation
- Added Functionality (if needed)

# Nuggets and Coal

- Nuggets
  - Allows embedding into the environment
  - It Works
  - Pretty easy to use (I think)
- Coal
  - Performance costs?
  - Library correctness?
    - Works for quake2 but is it correct?
  - Documentation
  - Other Platforms?