

# Python API Description Generator

Leah Chatkeonopadol and Smruthi Manjunath



# Overview

- ◆ API Descriptions
- ◆ Project Description
- ◆ Implementation
- ◆ Demo
- ◆ Summary

# API Descriptions

- Provide an overview of different services offered in web apps
- Makes it easier for developers to write code that interfaces with existing web apps

- API has a name
- API has at least one base URL
- API has at least one resource
- Each resource has at least one operation
- Each operation has a HTTP method
- There are no references to undefined types
- There are no references to undefined input bindings

# Project Description

- ◆ API description generator
  - ◆ Written for Google App Engine web apps
  - ◆ Python 2.7.3
  - ◆ Uses a combination of dynamic parsing (Python decorators) and static parsing

# Implementation

- ◆ Python decorators
  - ◆ Methods or classes that allow modification of code in functions or classes
  - ◆ Identified with a '@'
  - ◆ Takes an instance of a method or class object and returns the (possibly modified) object
  - ◆ Our decorator identifies method and class information and parses the given docstring

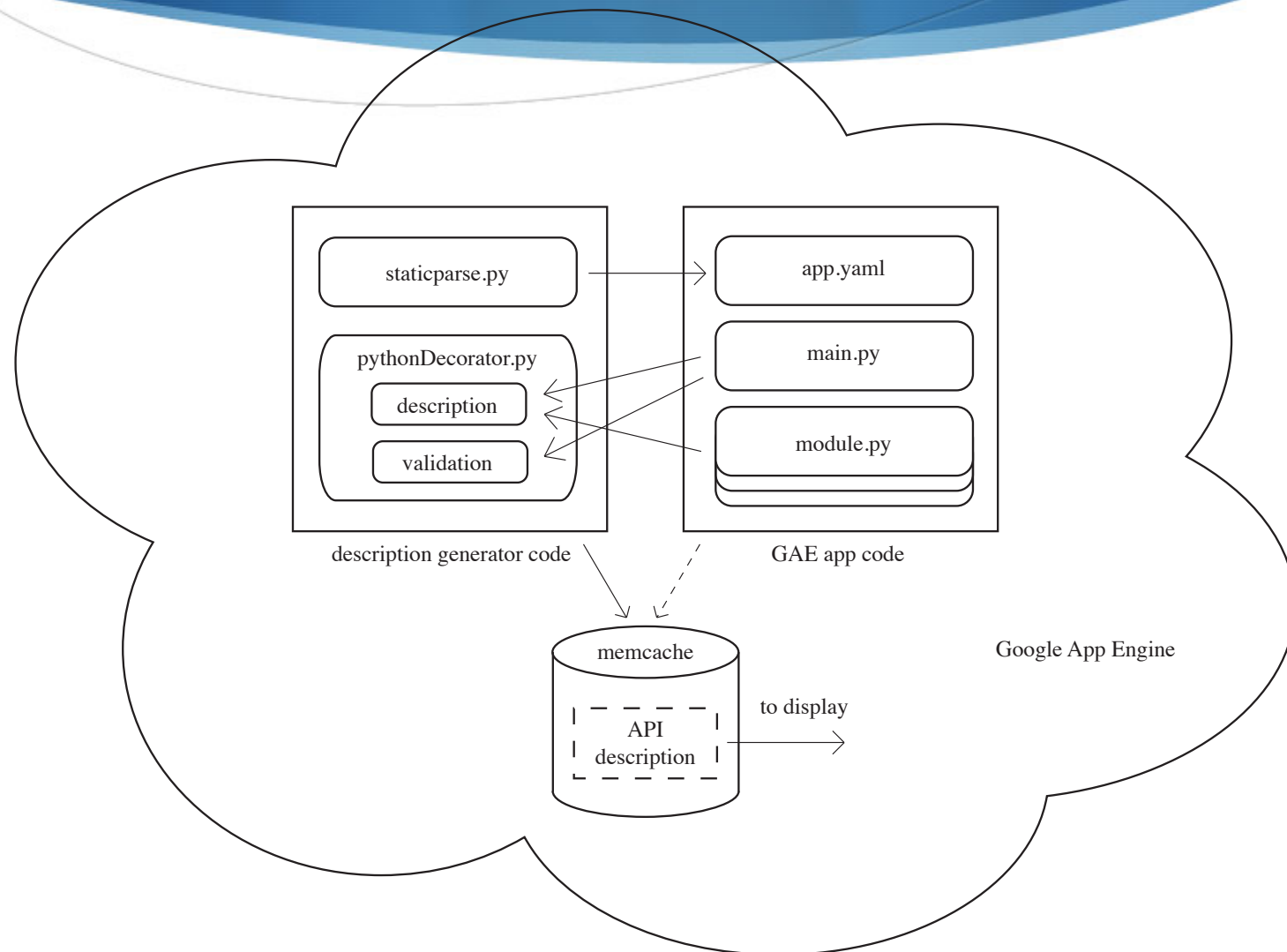
# Implementation

- ◆ Static parsing
  - ◆ Takes advantage of the format of the app.yaml file
  - ◆ Parses for the name, description, version, etc. of the application
  - ◆ Done once the first time the decorator is called

# Implementation

- ◆ Dynamic parsing
  - ◆ Docstrings obtained from the method or class object passed to the decorator
  - ◆ In a (method-less) class, parses for attributes
  - ◆ In an HTTP method, parses for arguments, input bindings, return types, and exceptions
  - ◆ Generates resources (classes of methods), operations (HTTP methods), and data types (method-less classes)

# Service Architecture





# Implementation

- ◆ API description stored in a dictionary in memcache
  - ◆ Expires in 1 hour
  - ◆ Because description is generated dynamically as each method is invoked, developer must click through all logical paths
  - ◆ Description retrieved by adding '/options' to base URL
    - ◆ If valid, converted to a JSON object and displayed as a web page
    - ◆ If invalid, displays error message(s)

Demo



# Summary

- ◆ Challenges faced
  - ◆ Python—dynamically typed
  - ◆ Use of decorators
  - ◆ Lack of standards for API descriptions
- ◆ Future work
  - ◆ Obtain more information about API
  - ◆ Extend to support non-GAE apps
  - ◆ Automate testing

# References

- ◆ Google App Engine: <https://developers.google.com/appengine/docs/python/overview>
- ◆ webapp2 framework: <http://webapp-improved.appspot.com>
- ◆ Python 2.7: <http://www.python.org/doc>
- ◆ Source code: <https://github.com/chatkeon/cs263>
- ◆ Project documentation: <https://apidescription.readthedocs.org/en/latest>

Thank You