

# **LIFF JS SDK Development Team Introduction**

# Shingo Sato / @sugarshin

Software Developer

LINE Frontend Development Center Team7

LIFF JavaScript SDK Team

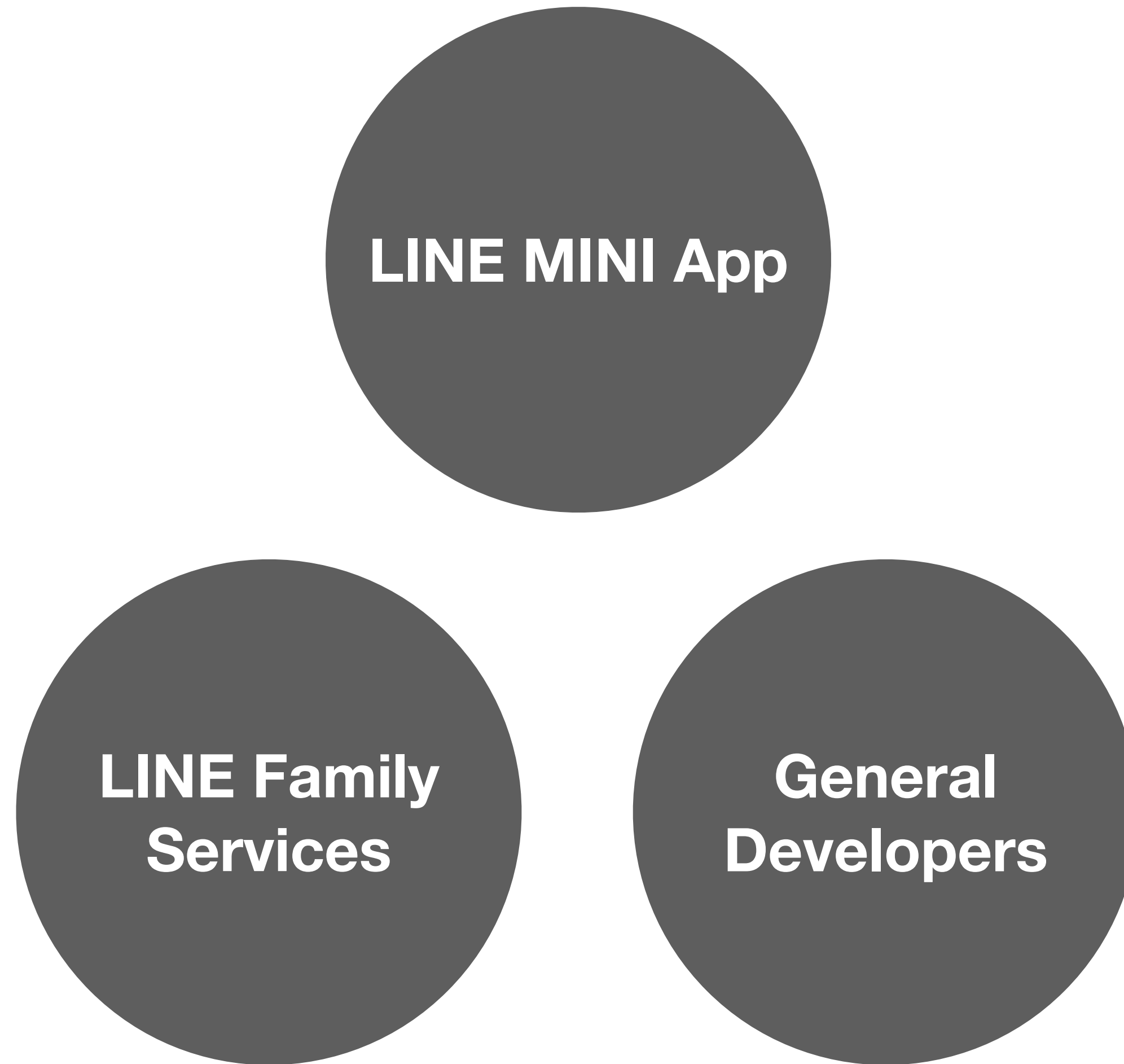


LINE

# WHAT IS LIF

LINE

# WHAT IS LIFF



# WHAT IS LIFF

Web application development platform

JavaScript, HTML, and CSS

Integration with LINE platform

Cross browser

LINE



# WHAT IS LIFF

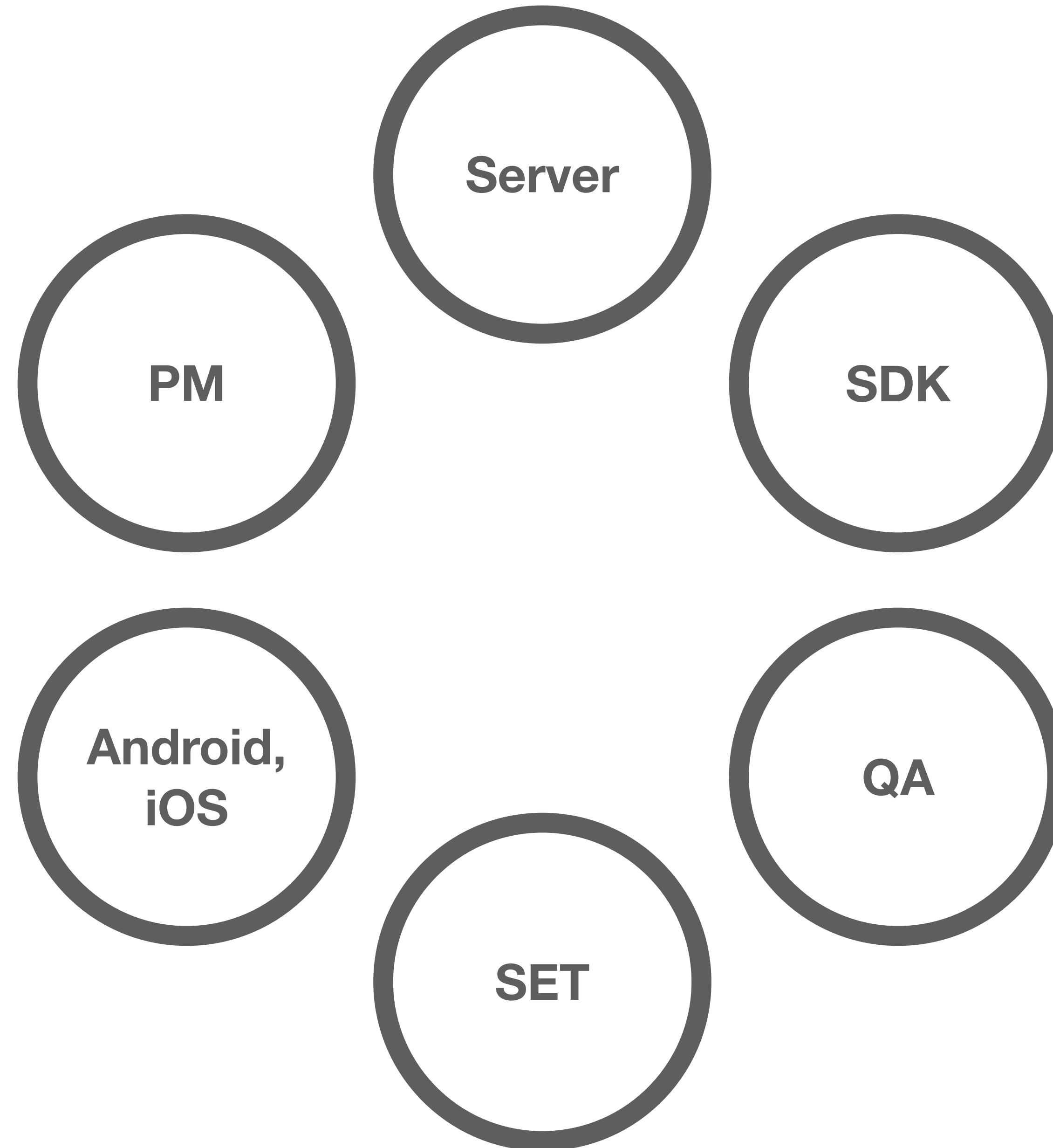
```
1  import liff from '@line/liff';
2
3  await liff.init({ liffId: 'yourLiffId' });
4
5  if (liff.isLoggedIn()) {
6    liff.getProfile();
7  } else {
8    liff.login();
9  }
10
```

# TEAM

LINE

# TEAM

## LIFF Teams

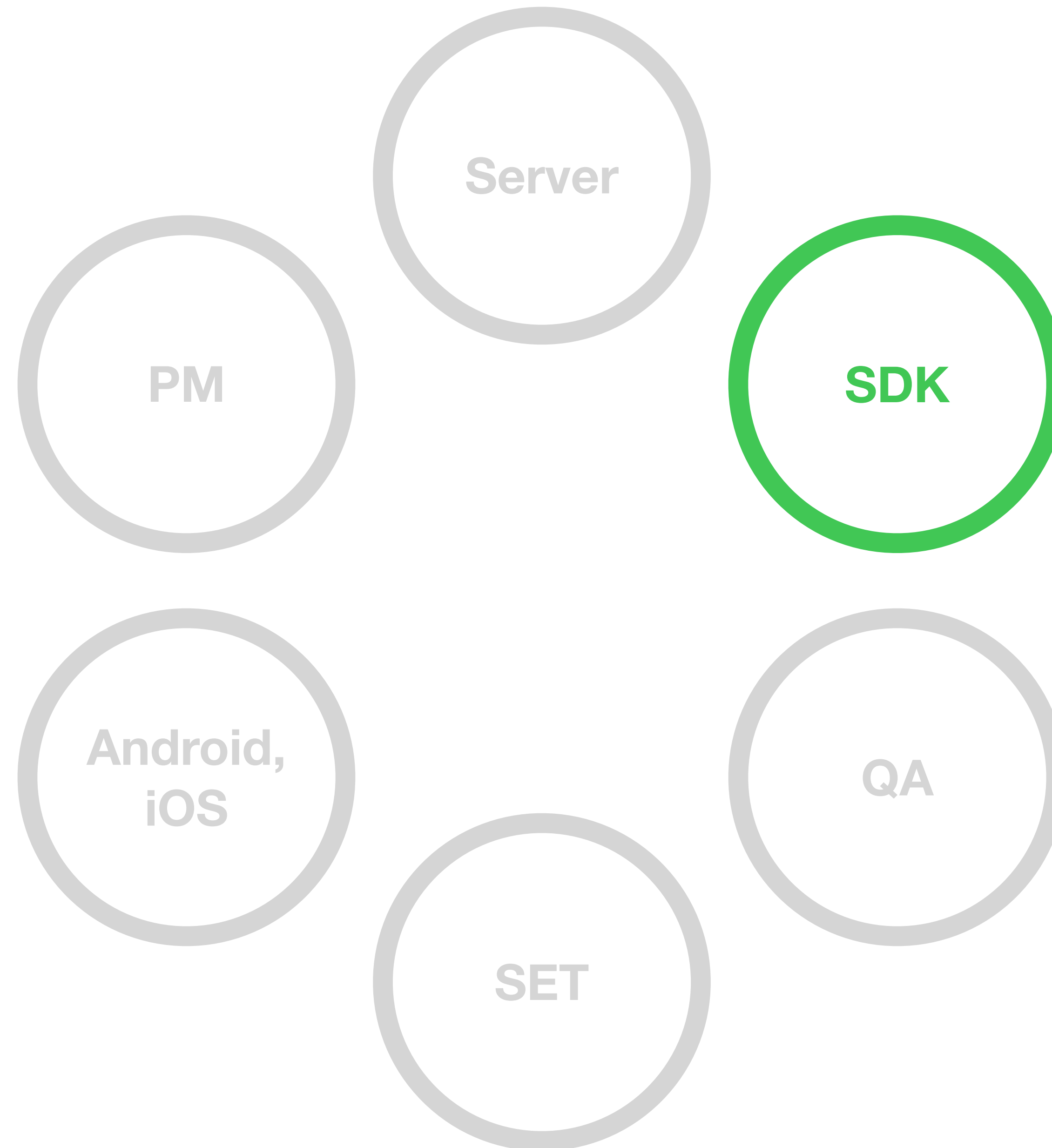


LINE



# TEAM

## LIFF Teams



LINE

# TEAM

## SDK Development Team



LINE

# TEAM

## SDK Development Team



Kyoto

Tokyo

LINE

# TEAM

## SDK Development Team Components

JavaScript  
SDK

LIFF Apps

Login

UI  
Components

Starter /  
Playground

Tools

LINE

# TEAM

## SDK Development Team Components

JavaScript  
SDK

LIFF Apps

Login

UI  
Components

Starter /  
Playground

Tools

LINE

# TEAM

## Architecture Review / Design Review

## Background and Issues

## Objective

## Goal

## Solution Ideas

## System Architecture, Implementation Design, I/F, etc.

LINE

Design doc - Mock LIFF server

作成者: kuroki@linecorp 最終更新: 2021 10 16 最終更新者: kuroki

1. Objective

2. Background

3. Problem statement

4. Solution

5. Goal

6. In scope

7. Out of scope

8. High-level design

9. Components

10. Proof of concept

11. Open issues

12. Appendix

Objective

According to the "Design doc - LIFF SDK / LINE native client integration test platform", we'll develop the mock LIFF server to reduce the point to cause test flakiness for the integration testing.

LIFF SDK Design Doc - Monorepo

See Design Doc 仕様書 (LIFF SDK / LINE native client integration test platform) 仕様書 (LIFF SDK / LINE native client integration test platform)

Monorepo for liiff-sdk

Issue and Background

1. liiff-sdk has low maintainability, scalability

- Not appropriate module dependencies and dependency injection
- Onset potential bugs
- Difficult to refactoring
- Difficult to develop a new feature

2. We plan to make the liiff-sdk to internal OSS

- It cannot easy to understand the structure/dependencies of liiff-sdk for developers outside the LIFF SDK team

Main purpose

- Reduce maintenance/develop new features costs through a properly encapsulated structure and the dependencies clarify
  - It also leads to improved internal/external quality
- Make the repository easy to understand structure and dependencies for developers outside the LIFF team to easily contribute to the liiff-sdk

Why Monorepo

Monorepo has the following advantages:

- Split modules easily
- Manage dependencies easily
- Consistent developer experience

And, Monorepo is also a standard way for managing/making a medium/large scale feature-rich library. So it's easy to understand for new developers and developers outside the LIFF team. Moreover, the playground, test apps, and tools for liiff-sdk can be developed on the same repository for code sharing and deploy/release unified.

PoC

please see: [shingo-sato/monorepo-testing](#)

Tool

Use Lerna and Yarn Workspaces

- Lerna
- Yarn Workspaces

ref:

- JavaScript Monorepo Tooling
- Why Lerna and Yarn Workspaces is a Perfect Match for Building Mono-Repo - A Close Look at Features and Performance

Directory structure

```
├── README.md
├── lerna.json
├── package.json
├── packages
│   ├── liiff
│   │   ├── packages-a
│   │   │   ├── README.md
│   │   │   ├── src
│   │   │   └── package.json
│   │   ├── packages-b
│   │   │   ├── README.md
│   │   │   ├── src
│   │   │   └── package.json
│   │   └── ...
│   └── ...
├── lerna
├── liiff
│   ├── README.md
│   ├── package.json
│   └── src
└── yarn.lock
```

Strategy of publish

- Publish with the "from-package" [lerna](#) option
  - > Publish packages in the latest commit where the version is not present in the registry
- Publish at [github](#) and [github/liiff](#) packages with bumped-up version even if only a small change in one package

functional

variation from the

works to cause test

doesn't work,

it is done by the

LIFF SDK works.

all the native apps

SDK initialization

diagram illustrates

# TECH STACK

LINE

# TECH STACK



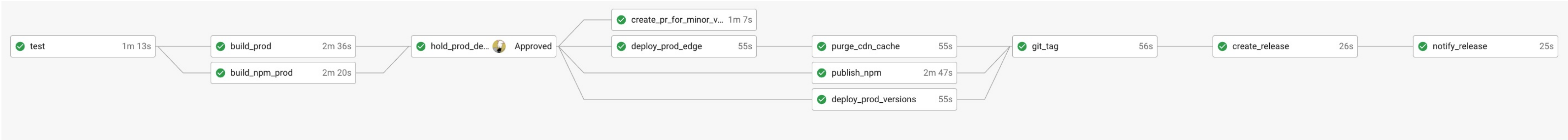
Lerna



LINE



# TECH STACK



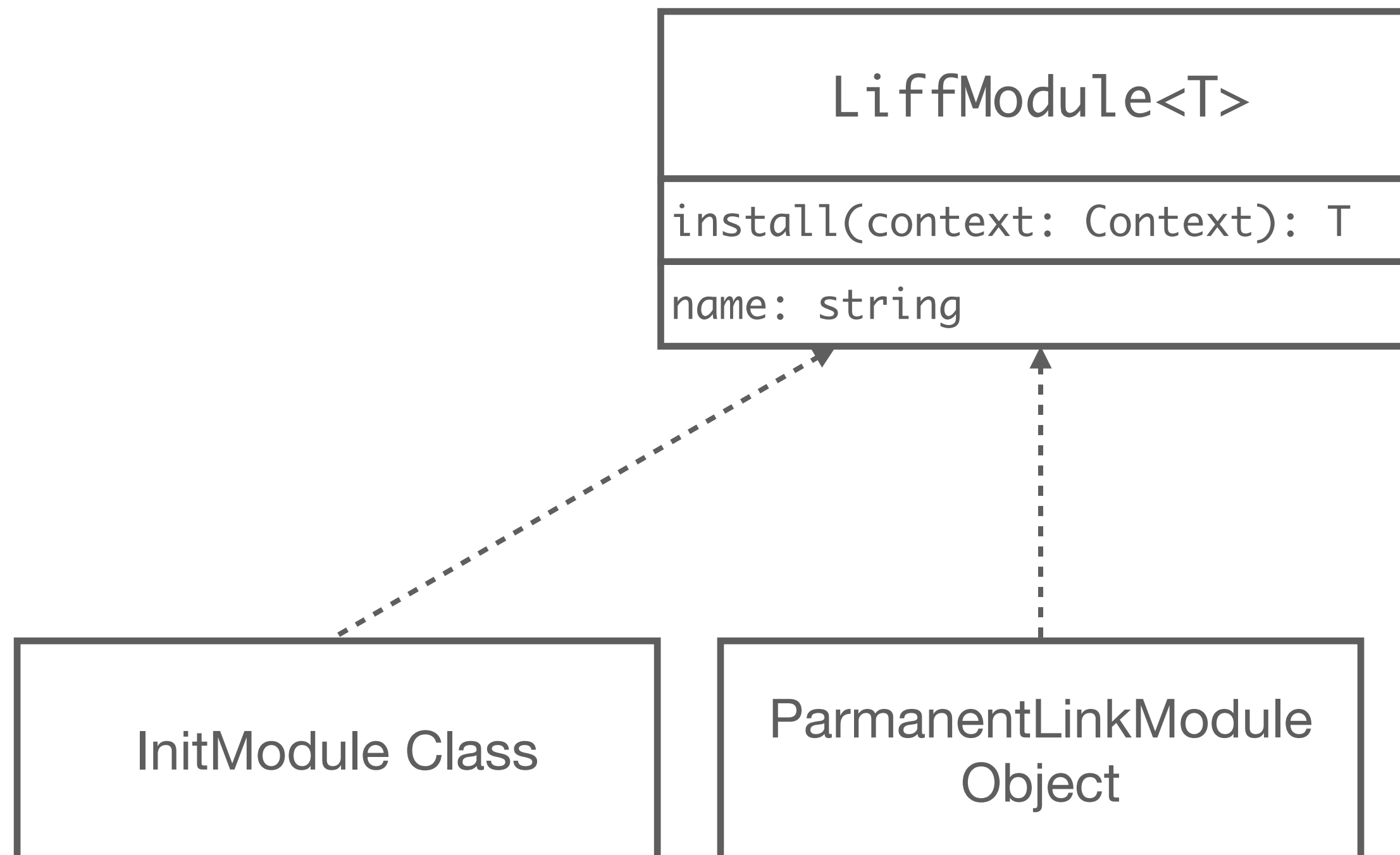
# ARCHITECTURE

LINE

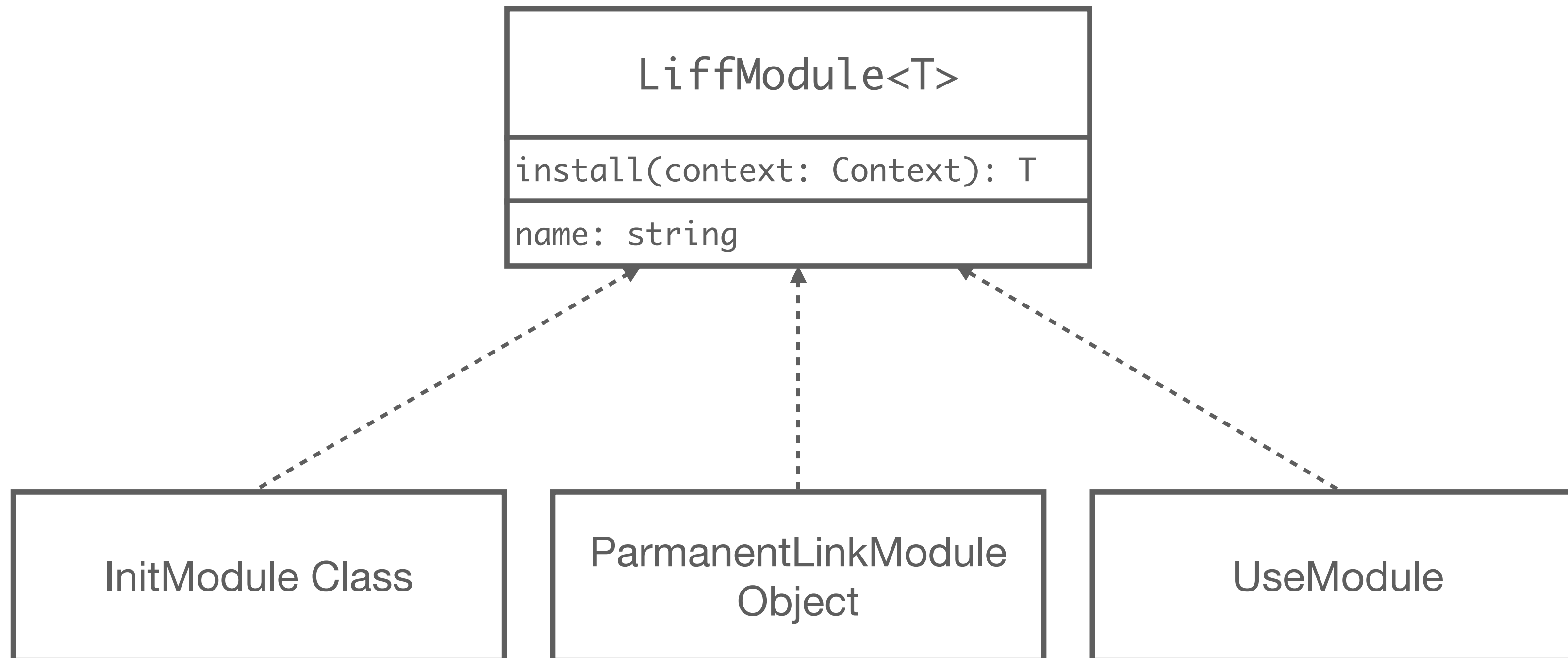
# ARCHITECTURE

LiffModule<T>
install(context: Context): T
name: string

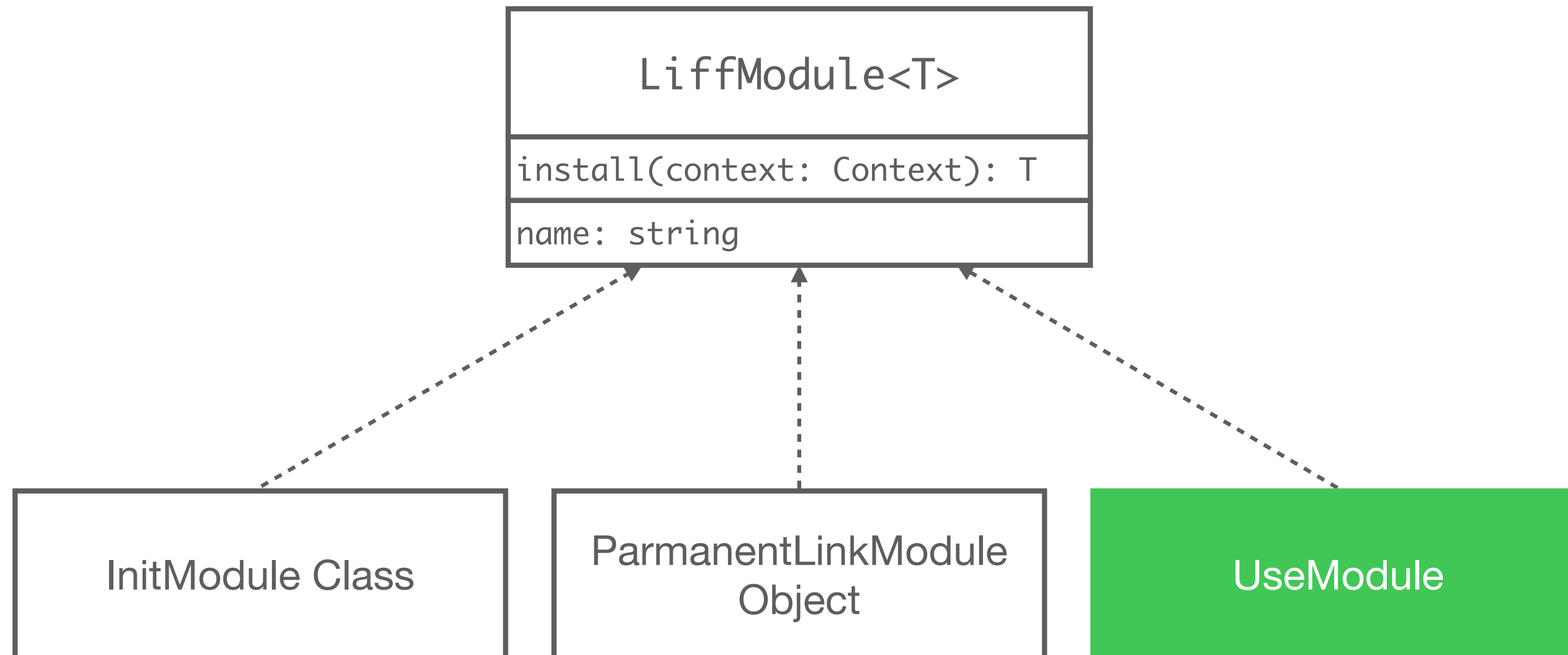
# ARCHITECTURE



# ARCHITECTURE



# ARCHITECTURE



# CHARACTERISTICS

LINE

# CHARACTERISTICS

Huge user base

LINE



# CHARACTERISTICS

Huge user base

Focus on platform development

LINE

# CHARACTERISTICS

Huge user base

Focus on platform development

Multi-knowledge

LINE

# CHALLENGE

LINE

# CHALLENGE

Developer Experience

LINE

# CHALLENGE

Developer Experience

Value Creation

LINE

# CHALLENGE

Value Creation

OSS

LINE

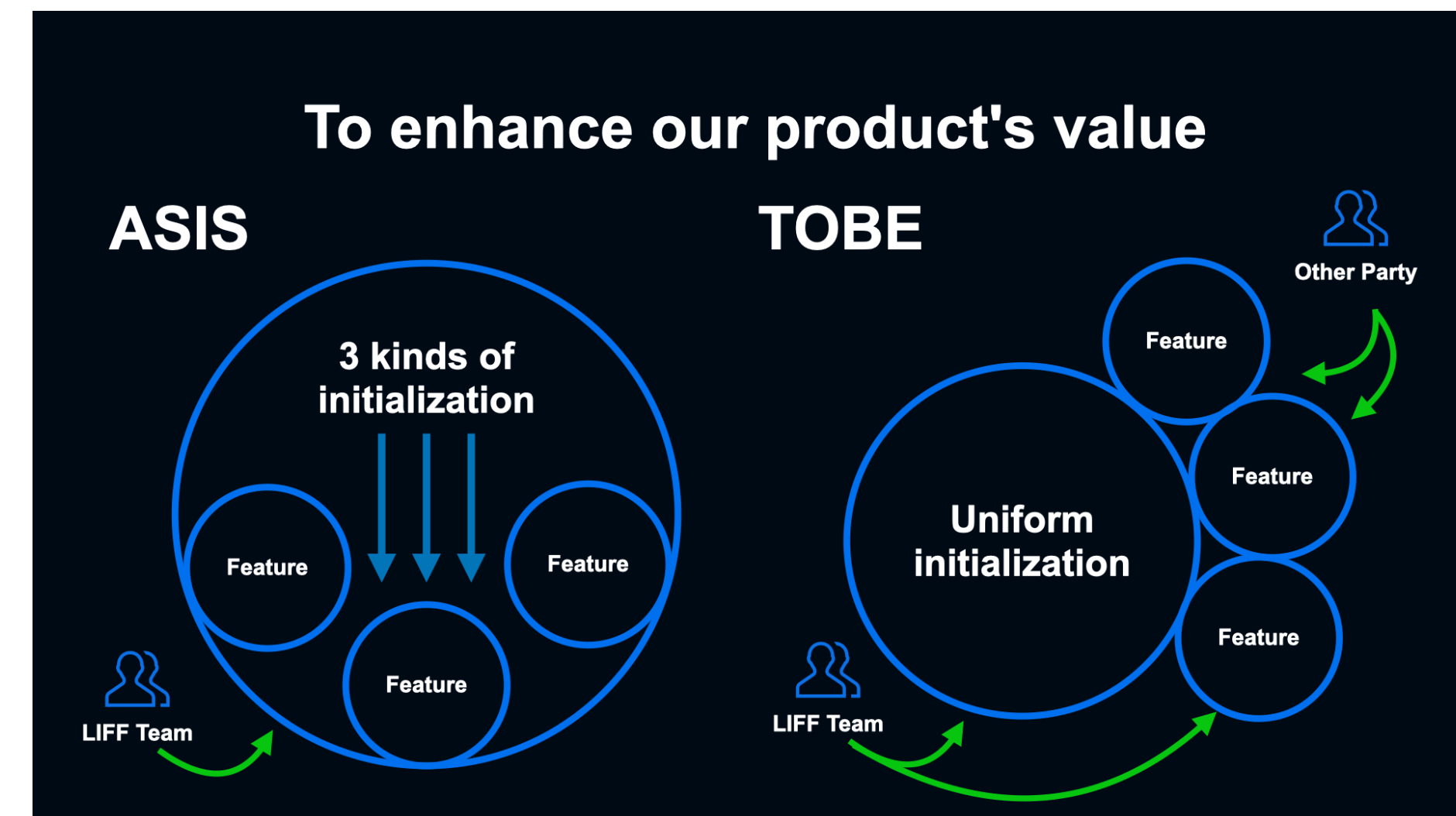
# CHALLENGE

Value Creation

OSS

Plug-in system

And so on



LINE

**THANK YOU**

LINE