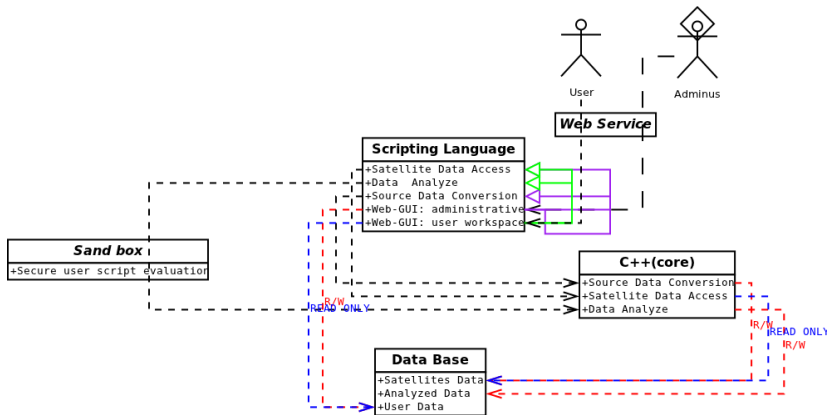


# Promis - проектування та огляд інструментарію

15 листопада 2011 р.

# Взаємодія програмних КОМПОНЕНТІВ



# Source Data Conversion

- ▶ Опис: Перетворення сировинних даних супутника та надає можливість їх збереження.
- ▶ Мета: Забезпечення інструментарію написання парсерів та конверторів даних для нових космічних проектів.

# Satellite Data Access

- ▶ Опис: Отримання та фільтрування даних з супутникових проектів.
- ▶ Мета: Забезпечення простого інтерфейсу доступу до даних супутникових проектів.

# Data Analyze

- ▶ Опис: Пакет математичних функцій та перетворень над масивами супутникових вимірів.
- ▶ Мета: Забезпечення аналізу, відображення та візуалізації даних.

# Source Data Conversion

- ▶ Опис: Перетворення та збереження сирцевих даних супутникових проектів.
- ▶ Мета: Забезпечення перетворення та збереження даних супутникових проектів. Обмеження конкретного парсеру збереженням даних саме цього супутникового проекту.

# Satellite Data Access

- ▶ Опис: Доступ до даних супутникових проєктів з обмеженням лише на читання.
- ▶ Мета: Надання зручного доступу та фільтрації даних без можливості зміни їх джерела.



# Data Analyze

- ▶ Опис: Обгортка C++ модулю аналізу даних, що забезпечує ґручний і зрозумілий потік даних крізь аналізатори.
- ▶ Мета: Побудова зручного API для аналізу даних та генерації простих скриптів у графічному режимі.
- ▶ Нагальні задачі:
  - ▶ Зручна модель представлення даних.
  - ▶ Застосування Method Chaining або Fluent Interface, як моделі оперування даними.

# Web-GUI (administrative)

- ▶ Опис: Частина користувацького інтерфейсу, що забезпечує керування користувацькими даними, супутниковими проектами, процесом постачання та аналізу даних.
- ▶ Мета: Створення керівного модулю.
- ▶ Функції керування:
  - ▶ Користувацькими даними/запитами.
  - ▶ Супутниковими проектами.
  - ▶ Процесом обробки супутникових проектів.

# Web-GUI (administrative)

- ▶ Опис: Частина користувацького інтерфейсу, що представляє доступ до статичної інформації про проект та до робочого(дослідницького) простору.
- ▶ Мета: Забезпечення доступу до матеріалів проекту.

# Засоби реалізації поставлених задач

- ▶ C++: STL/Boost/GSL Написання наукомісткої частини, що включатиме фільтри, функції та методи аналізу, стандартні алгоритми.
- ▶ PostgreSQL База даних, центральне сховище супутникових даних, тимчасових та користувацьких даних.
- ▶ Gearman Контроль та розподіл виконання завдань між кількома потоками/серверами.
- ▶ JS: JQuery Інтерактивна взаємодія з користувачем на клієнтській стороні.

# Скриптова мова

Основна проблема, що виникає при виборі - безпечне виконання на стороні серверу. Варіанти.

- ▶ **Python / Virtualization** Можливість безпечного запуску сторонніх скриптів є небажанняю з версії 2.6 і вимкнена в 3.0 через проблеми безпеки.
- ▶ **JavaScript / Node.js** Можливе виконання на серверній стороні за допомогою Node.js, що працює на базі v8(Google). Безпечне виконання на базі налаштувань та обмежень Node.js.

# Наслідки використання

- ▶ Python / Virtualization Використання Django та супунього серверу для побудови веб-інтерфейсу. Підняття віртуальних машини для виконання скриптів.
- ▶ JavaScript / Node.js Використання PHP / Yii для написання веб-інтерфейсу. Підняття Apache / nginx серверів для виконання PHP.

# Документування



# Розробник-розробнику

Має вигляд:

- ▶ Набір UML схем, що спростять розуміння архітектури проекту, та певних складних або вузьких елементів програмної реалізації чи алгоритмів.
- ▶ Вікі-сторінки з описом та обговоренням ідей алгоритмів, реалізації.
- ▶ Система документації та обговорення багів.
- ▶ Система опису життя та розвитку проекту: мінорні та мажорні версії, milestones і т.п.

# Розробник-користувачу: API класів(Class Reference)

- ▶ Генерується з програмного коду за допомогою Doxygen.
- ▶ Містить опис усіх функцій та їх вхідни та вихідних параметрів.
- ▶ Містить важливі приклади використання.
- ▶ Коротко та просто.

# Розробник-користувачу: Ознайомлюючий ввід(Definitive guide)

- ▶ Докладний огляд усіх базових принципів та ідей за прикладами та ліричними відступами.
- ▶ Огляд ідей та ідіом, що варто використовувати при роботі з інструментарем проекту.
- ▶ Приклади розв'язку задач, що часто виникають в практиці.

# Користувач-користувачу: Користувацькі статті(Wiki pages)

- Опис розв'язку складних, нестандартних проблем.

## Нагальні питання

- ▶ Вибір скриптових мови та супутнього пакету проблем до кожної з них
- ▶ Створення списку математичних функцій, фільтрів що будуть реалізовуватись
- ▶ Створення типу даних для передачі та зручного трансферу та їх перетворення