

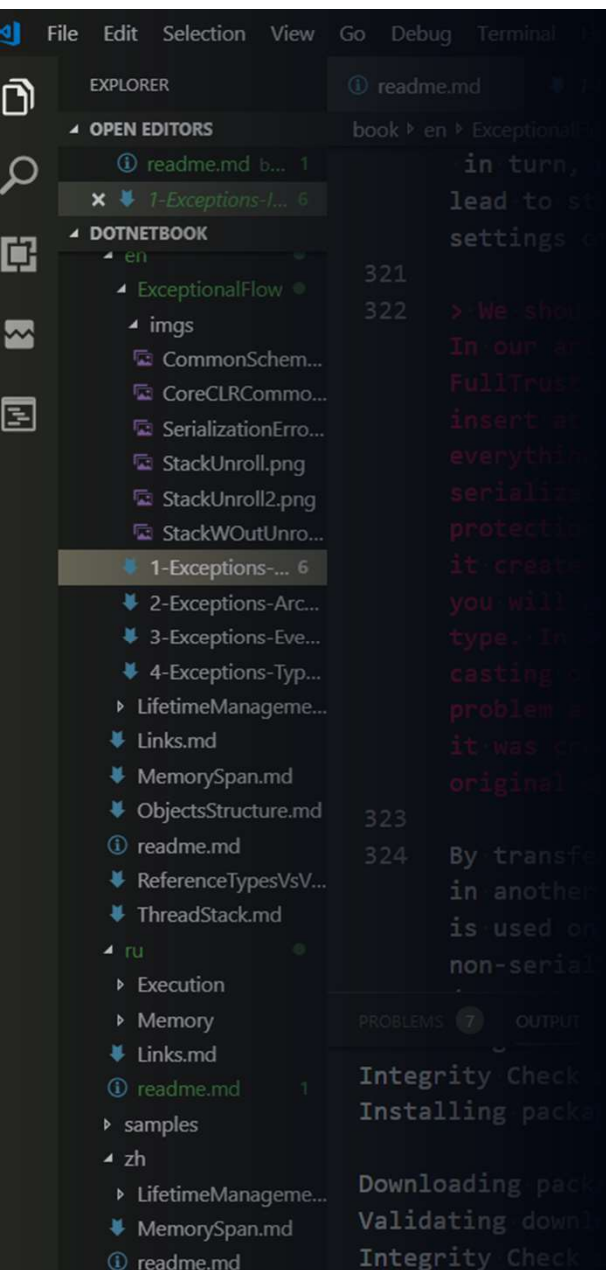
Плагины к микросервисам и Драйверы доступа к данным

Безопасное расширение функциональности



Сидристый Станислав





Станислав Сидристый

- Архитектор, руководитель направления
- Склад: gitverse.ru/sidristij/
- Запасной: github.com/sidristij/
- Для телеграмм: @sidristij
- Электронпочта: sunex.development@gmail.com

Решаемые проблемы

почему для решения задачи
могут быть выбраны плагины

Решаемые проблемы

- Интеграция с внешними системами:
 - Унификация доступа к данным и реализация доступа к данным через драйвер (один или несколько)
 - При корректной реализации нет необходимости поднимать по 1 микросервису на каждое решение
- Обработка специфичных данных
 - Сложные математические выражения
 - Дополнение языковой модели внутреннего языка запросов
 - Обработка дополнительных условий и выражений
- Решение задач предобработки, постобработки данных в точке, находящейся близко к данным
- Многократное повышение производительности решения.

Пример задачи

«потокотый онлайн-excel»

основной функционал

- Ячейка – это данные в некотором формате
 - Строка
 - Дата
 - Число
 - **Формула** + сложность вызова и приоритет для планирования + TTL кэша
- У ячейки есть координаты
 - Дата и время события
 - Название временного ряда
 - Некоторая группа принадлежности ряда (система,
- Основная библиотека функций содержит базовый функционал
 - sin/cos/базовая алгебра
 - Basic programming: if-else, do-while, while, переменные
 - Работа с временным рядом (дай ближайшую слева дату ячейки по переданной дате)
 - Конвертация типов для записи в ячейку
 - Стоимость вызова функции для оценки планирования

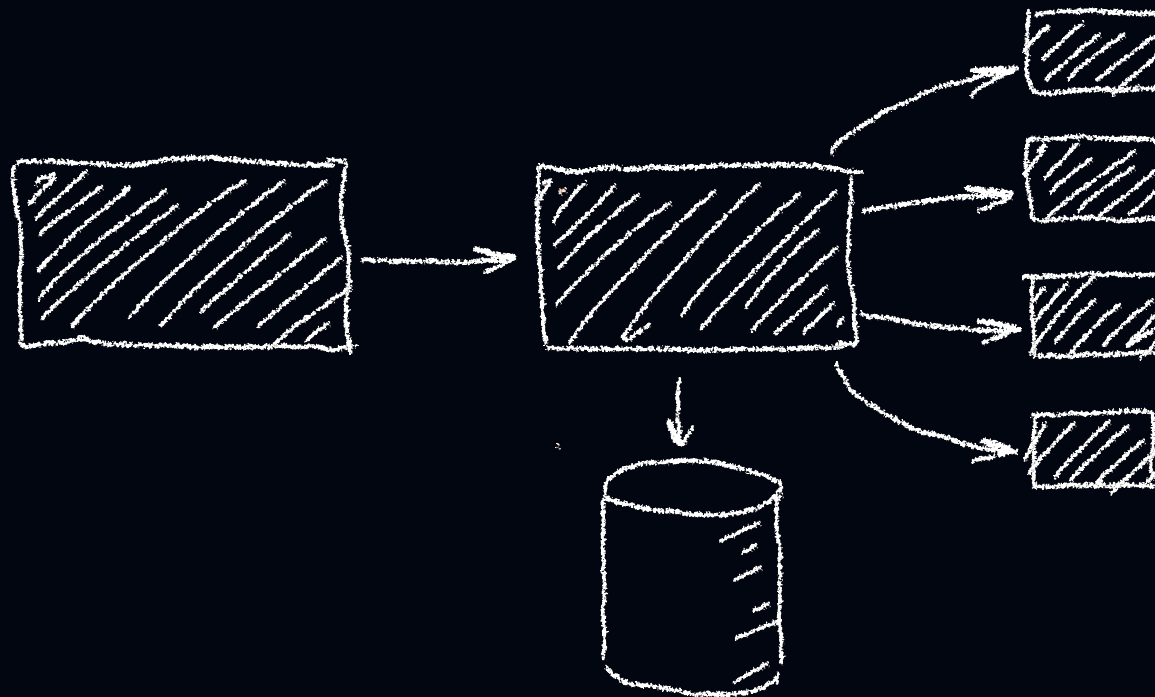
Пример задачи

«потокотый онлайн-excel»

дополнительный функционал, реализованный разными плагинами

- Ячейка – это данные в некотором формате
 - (+) новые типы данных и конвертации (например, bool, матрица)
- У ячейки есть координаты
 - (+) права доступа и проверка доступа
- Дополнительный функционал к библиотеке функций
 - (+) Интегралы, производные
 - (+) Работа с матрицами
 - (+) Вызов внешних систем (например, функция 1C-QUERY("...", TTL_запроса), которая обратится в Сервис доступа к 1С и считает оттуда данные

Варианты решения

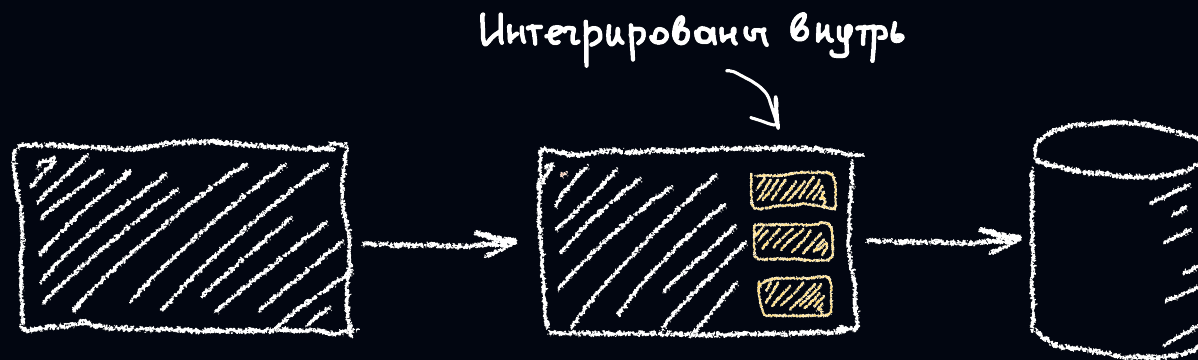


Плюсы:

- Независимое обновление каждой функции
- Полная изоляция каждой функции и её не влияние на центральный сервис расчётов

Минусы:

- Тяжёлая передача контекста делает вычисления многократно дороже их самих

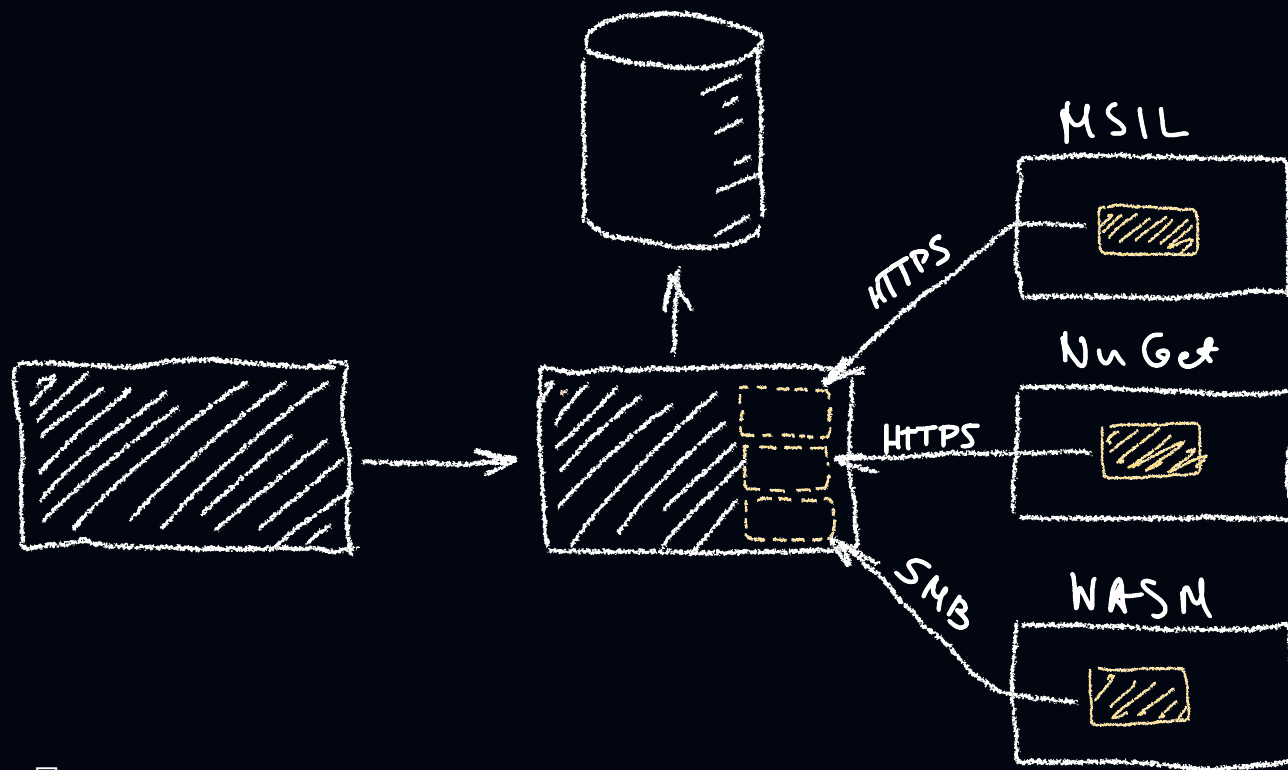


Плюсы:

- Максимально производительное решение
- Можно выстроить цепочки расчётов по зависимостям т.о. чтобы деревья зависимостей обсчитывались мгновенно
- Все зависимые данные можно забрать одним запросом, снизив нагрузку на БД

Минусы:

- Необходима изоляция функций т.о. чтобы они не влияли на сервис
- При обновлении плагинов надо обновлять сам сервис



Плюсы:

- Максимально производительное решение
- Можно выстроить цепочки расчётов по зависимостям т.о. чтобы деревья зависимостей обсчитывались мгновенно
- Все зависимые данные можно забрать одним запросом, снизив нагрузку на БД
- При обновлении плагинов центральный сервис обновлять не надо

Минусы:

- Необходима изоляция функций т.о. чтобы они не влияли на сервис

Расширения в dotnet core

Расширения в dotnet core

Assembly.LoadFrom

- + Очень просто сделать
- + Можно изолировать по DI
- Нет изоляции по версиям сборок
- Нельзя выгрузить и обновить

Новые процессы

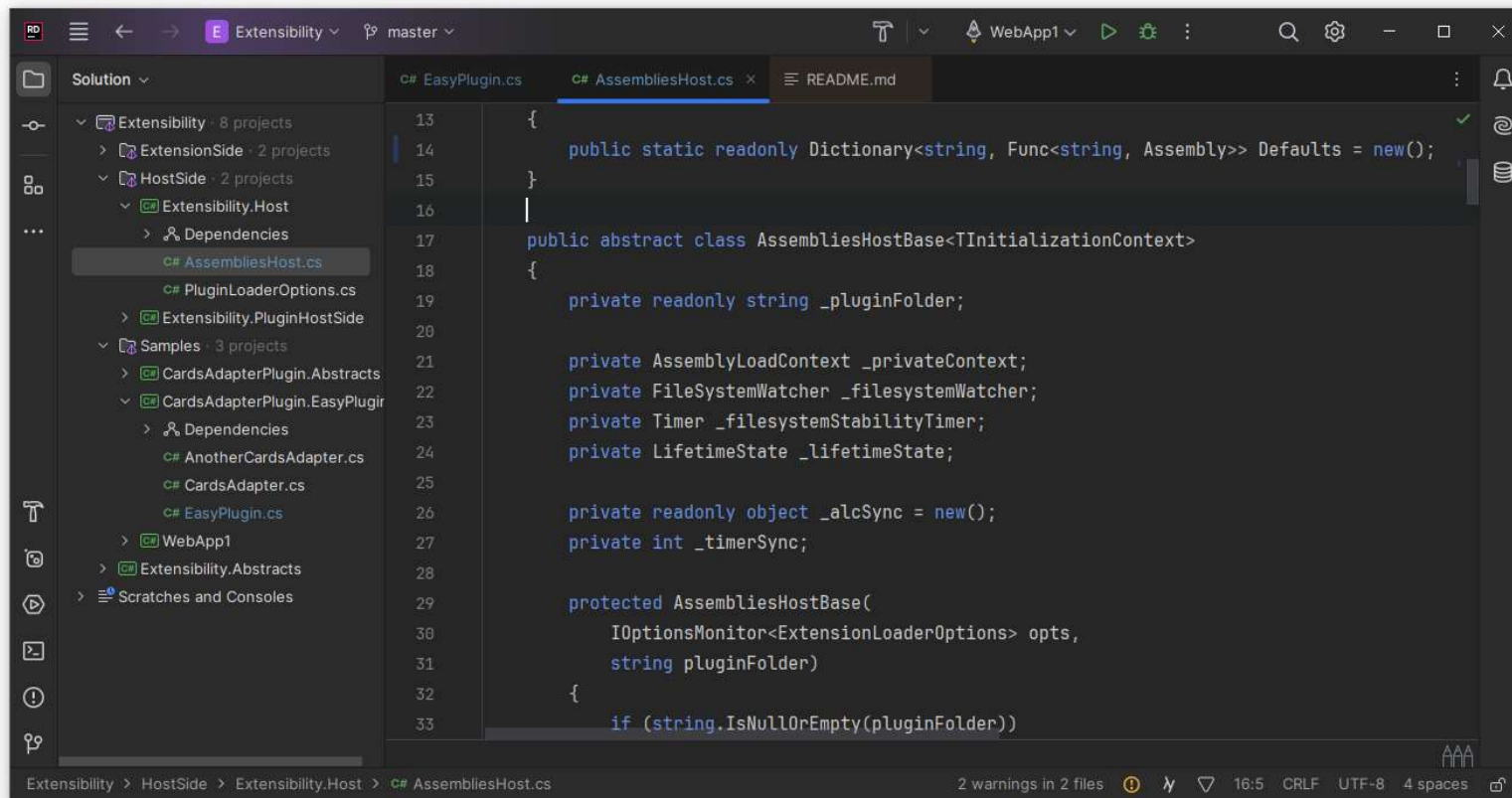
- + Полная изоляция по памяти
- Для взаимодействия нужен маршаллинг
- + Изоляция по версиям сборок
- Нельзя выгрузить и обновить

AssemblyLoadContext

- + Изоляция по памяти через DI + Castle
- + Для взаимодействия не нужен маршаллинг
- + Изоляция по версиям сборок
- + Можно выгрузить и обновить
- + Относительно просто сделать
- Ментально сложнее принять решение об использовании



Код



```
13 {
14     public static readonly Dictionary<string, Func<string, Assembly>> Defaults = new();
15 }
16 |
17 public abstract class AssembliesHostBase<TInitializationContext>
18 {
19     private readonly string _pluginFolder;
20
21     private AssemblyLoadContext _privateContext;
22     private FileSystemWatcher _filesystemWatcher;
23     private Timer _filesystemStabilityTimer;
24     private LifetimeState _lifetimeState;
25
26     private readonly object _alcSync = new();
27     private int _timerSync;
28
29     protected AssembliesHostBase(
30         IOptionMonitor<ExtensionLoaderOptions> opts,
31         string pluginFolder)
32     {
33         if (string.IsNullOrEmpty(pluginFolder))
```

Dockerfile

```
FROM yourservice:tag AS BUILD

#...
WORKDIR /app/product/plugins/
COPY . .

FROM aspnet AS RUN
#...
ENV PLUGINS_PATH=/app/product/plugins/
#...
```

либо SMB

99.99	99.50	99.00
Auth.	group	---

+

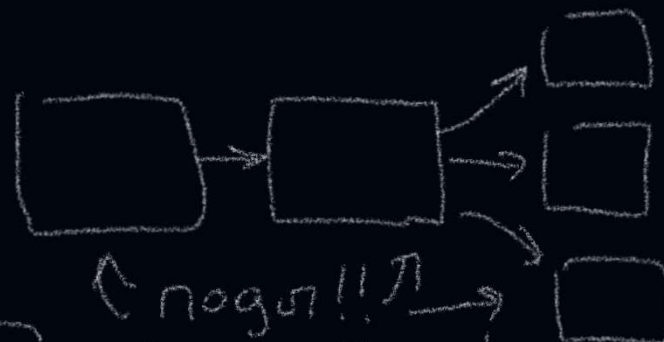
ПАРОЛЬ ОТ
СЕРВЕРНОЙ

9E56BA ~~9E56BA~~

↑
НИКОМУ НЕ
СООБЩАТЬ



ВЫВОДЫ



подоп!!л
без проблем!



QA