Національний університет «Києво-Могилянська академія»

РОЗРОБКА ТА ВПРОВАДЖЕННЯ МІКРОФРОНТЕНД-ФРЕЙМВОРКУ для ПОБУДОВИ МАСШТАБОВАНИХ ВЕБЗАСТОСУНКІВ

Підготував: Мисько Юрій
Науковий керівник: Нагірна А. М.

МЕТА РОБОТИ ТА ЗАВДАННЯ

Мета:

Розробка та впровадження мікрофронтенд-фреймворку

Завдання:

1. Адаптація принципів мікросервісної архітектури до умов браузерного середовища виконання.
2. Порівняльний аналіз існуючих рішень та формування вимог до фреймворку.
3. Спроєктувати архітектуру фреймворку на основі даних вимог.
4. Реалізація та апробація фреймворку.

ПРОБЛЕМИ МОНОЛІТНОГО ФРОНТЕНДУ

1. Експоненціальне зростання часу збірки проєкту
2. Складність налагодження у великих кодових базах
3. Конфлікти при паралельній роботі команд
4. Обмеження у виборі технологічного стеку
5. Складність поступових змін архітектури

ПРОБЛЕМИ ІСНУЮЧИХ РІШЕНЬ

1. Необхідність спеціалізованих адаптерів для кожного фреймворку користувацьких інтерфейсів.
2. Відсутність способів типобезпечної комунікації.
3. Складна ручна конфігурація.
4. Спільний стан застосунку — спільна точка відмови.
5. Невідповідність принципам мікросервісної архітектури.

ПЕРЕНЕСЕННЯ ПРИНЦИПІВ МІКРОСЕРВІСІВ

1.Незалежність розгортання.

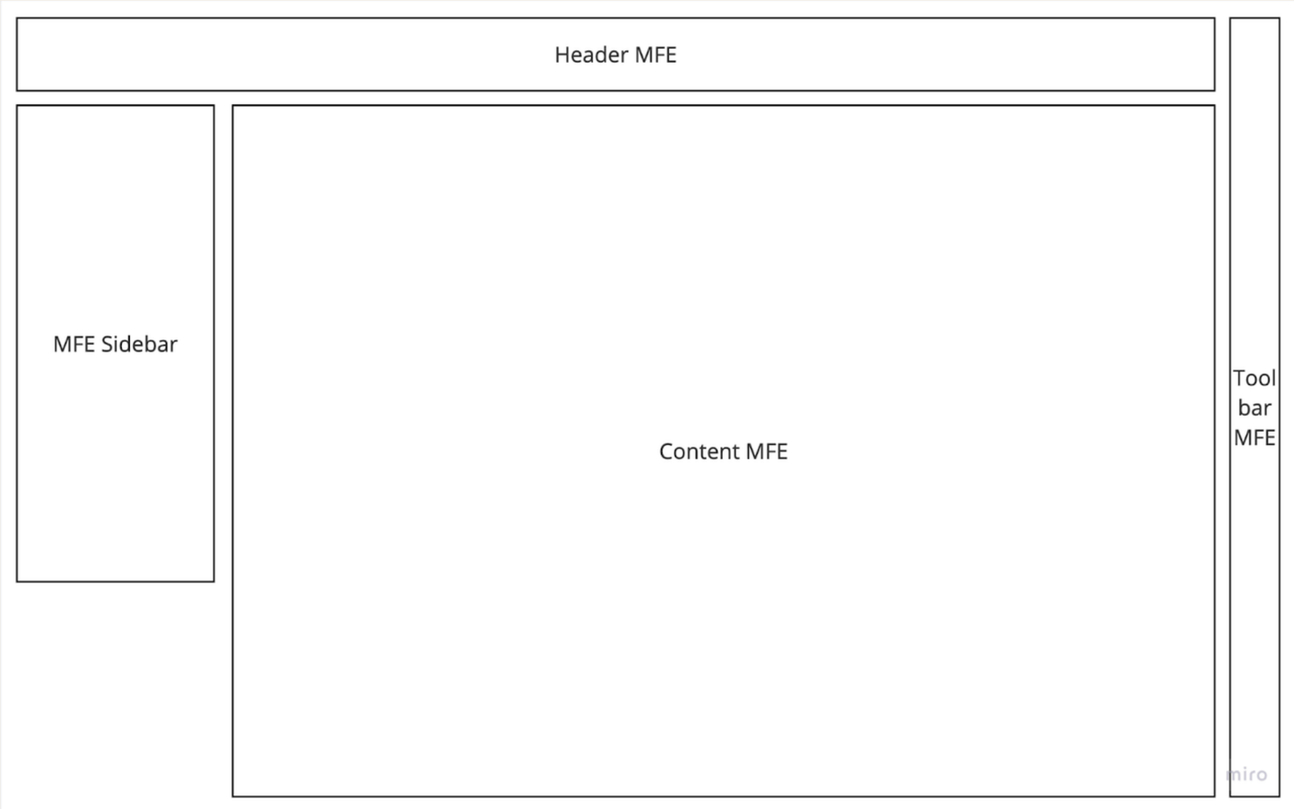
2.Організація навколо бізнес-домену.

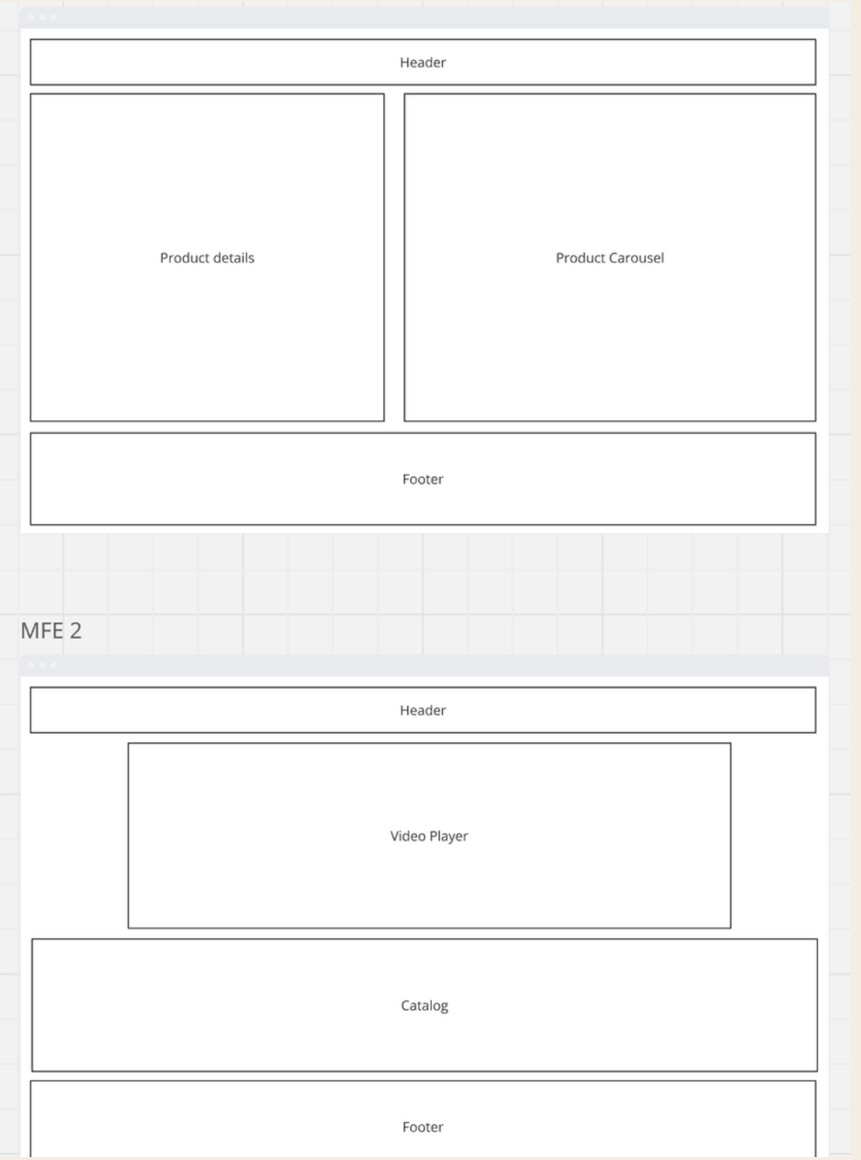
3.Децентралізоване керування даними.

4.Приховування деталей імплементації.

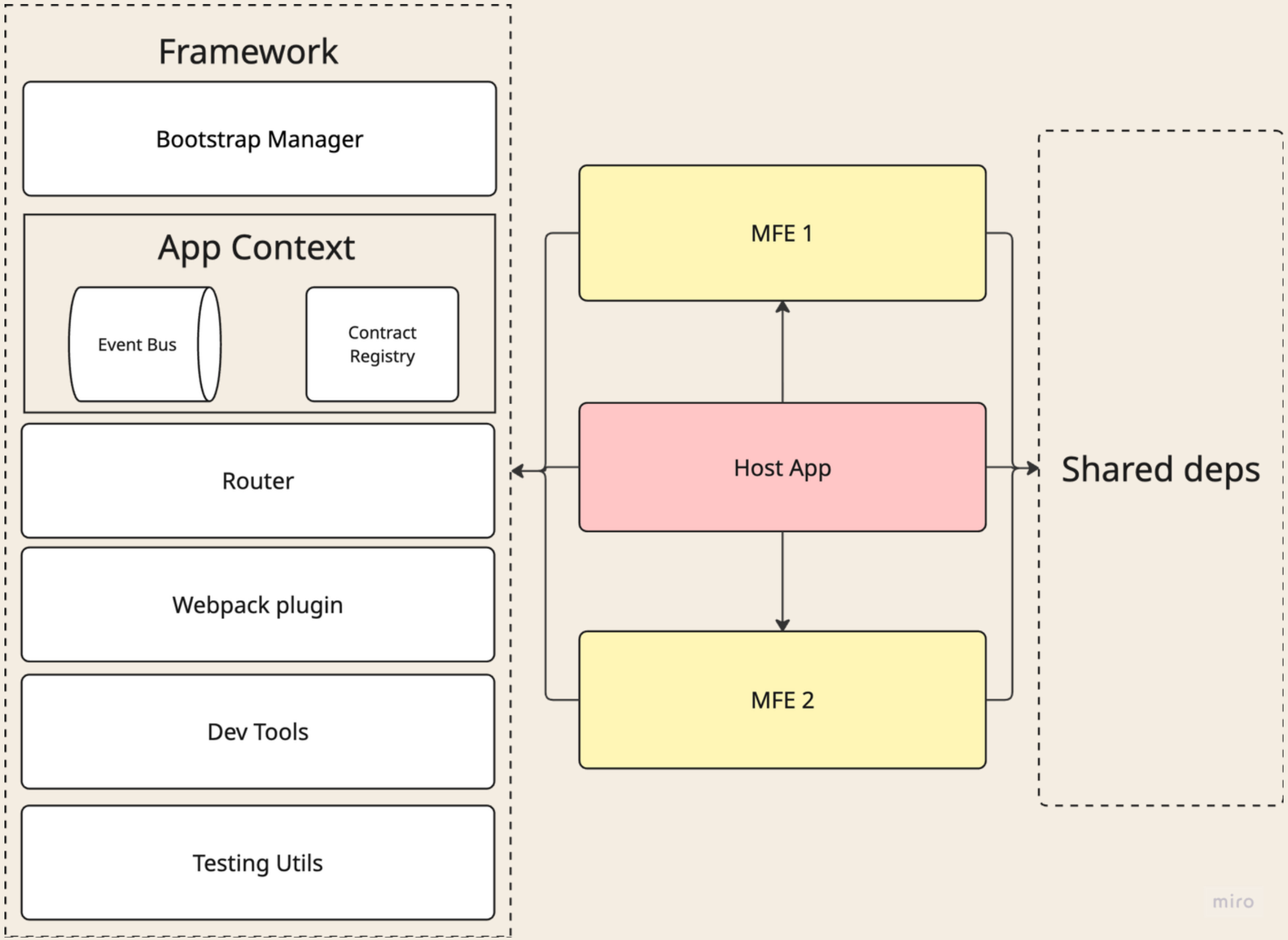
5.Ізоляція відмов.

ГІБРИДНИЙ ПІДХІД ДО ПОДІЛУ

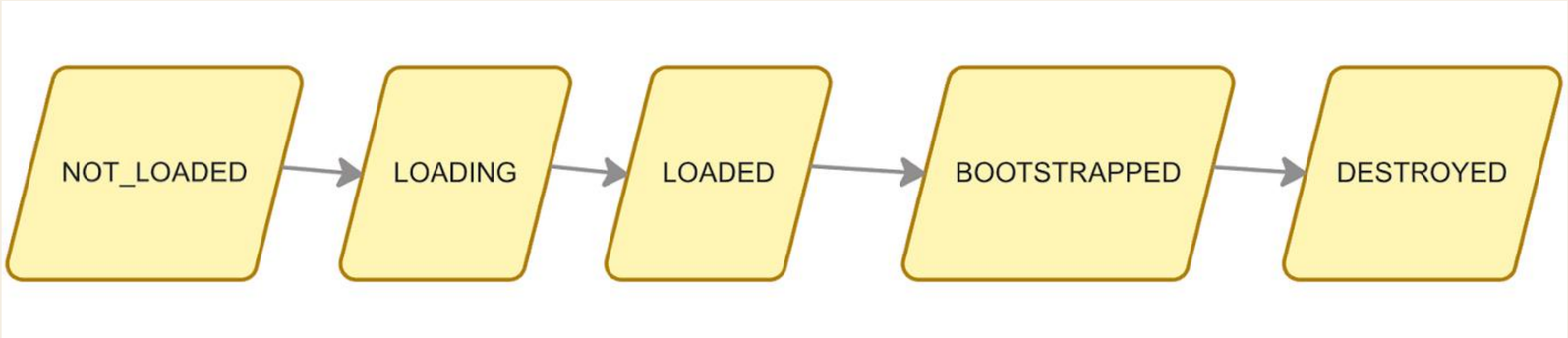




АРХІТЕКТУРА ФРЕЙМВОРКУ



BOOTSTRAP MANAGER



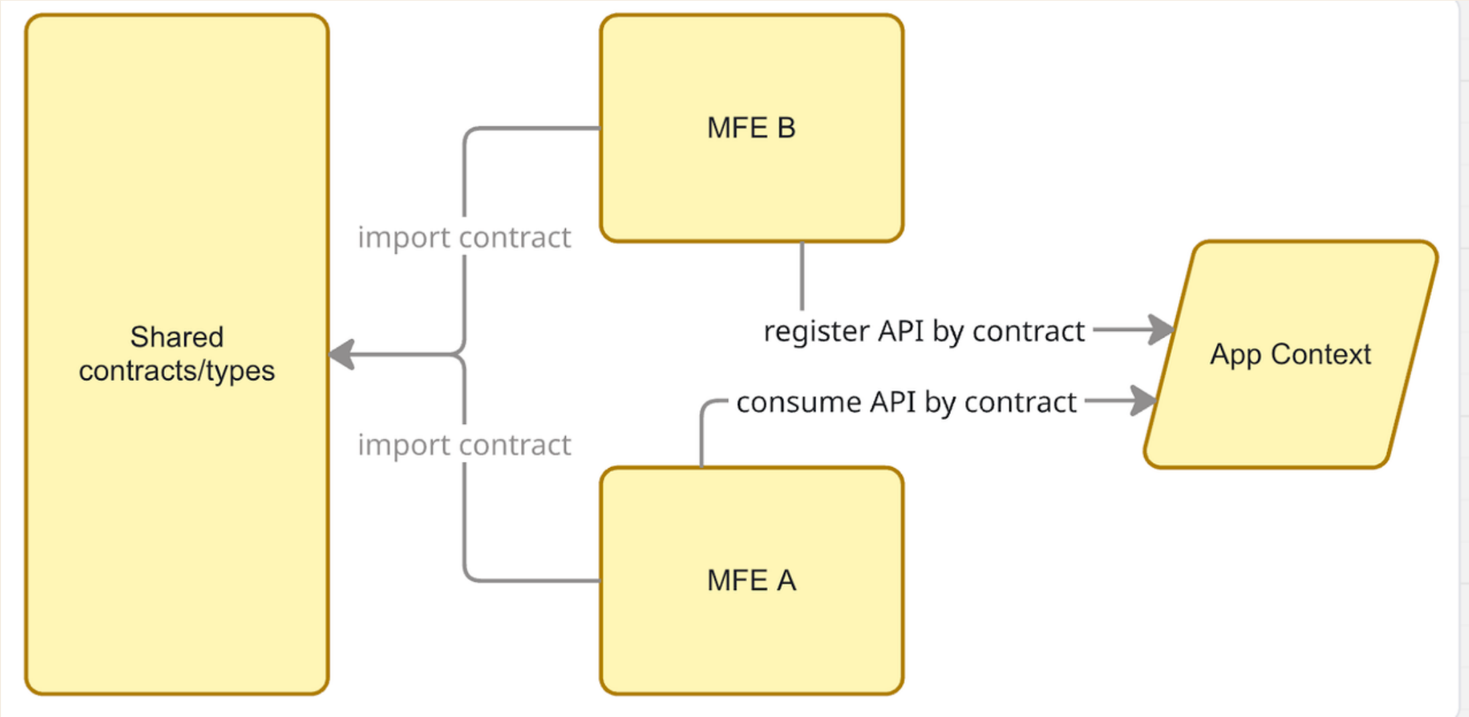
1.Активація мікрофронтендів за певної умови(зміна URL).

2.Топологічне сортування.

3.Підтримка пріоритету активації.

4.Керування життєвим циклом мікрофронтендів.

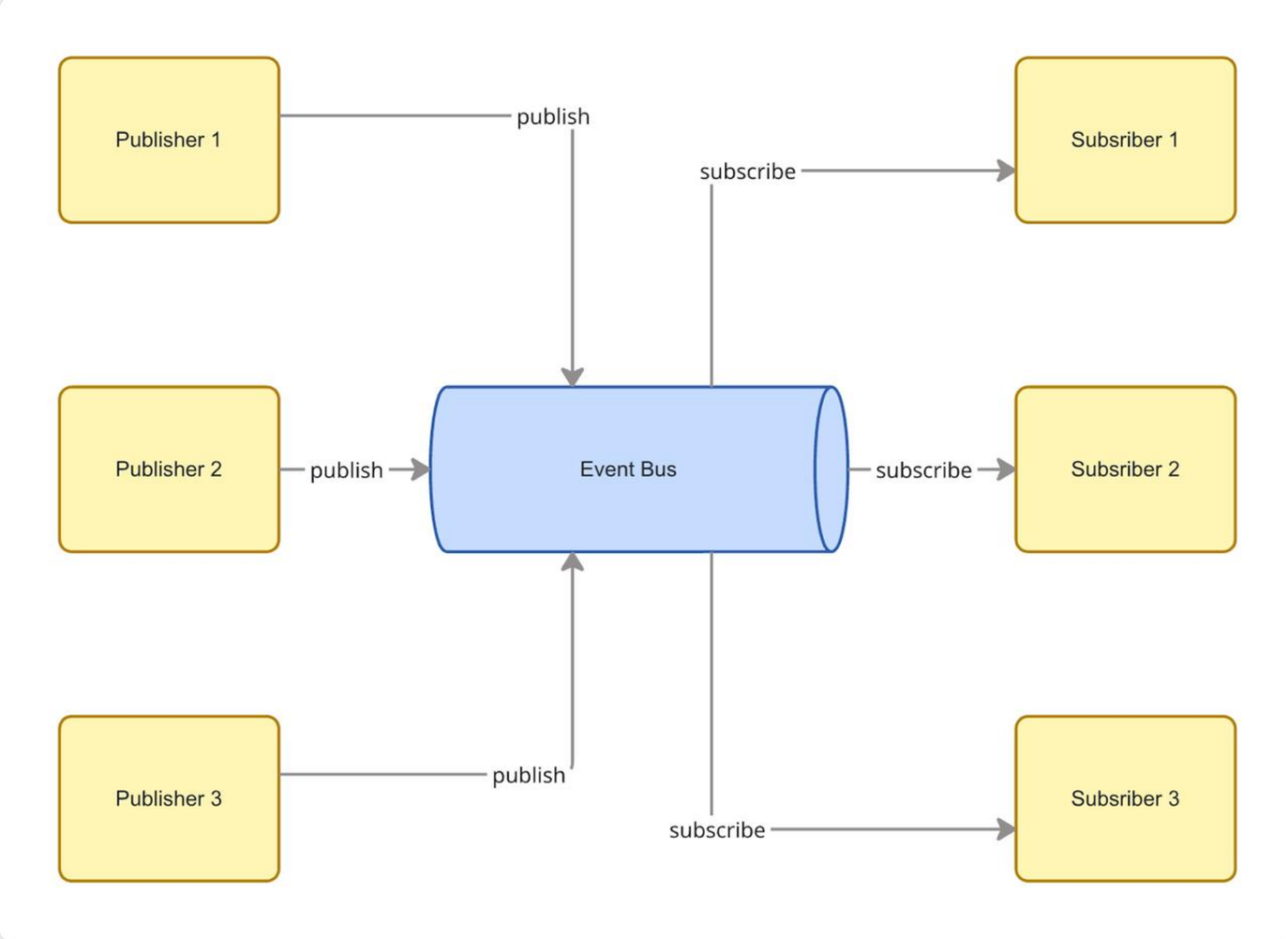
APP CONTEXT




```ts
export interface AppContext {
    register<T>(contract: Contract<T>, implementation: T): void;
    use<T>(contract: Contract<T>): T;

    publish<T>(eventContract: EventContract<T>, payload: T): void;
    subscribe<T>(eventContract: EventContract<T>, handler: (payload: T) => void): EventSubscription;
}
```

ШИНА ПОДІЙ


Publisher 1
publish
Subsriber 1
subscribe
Publisher 2
publish
Event Bus
subscribe
Subsriber 2
Publisher 3
publish
subscribe
Subsriber 3

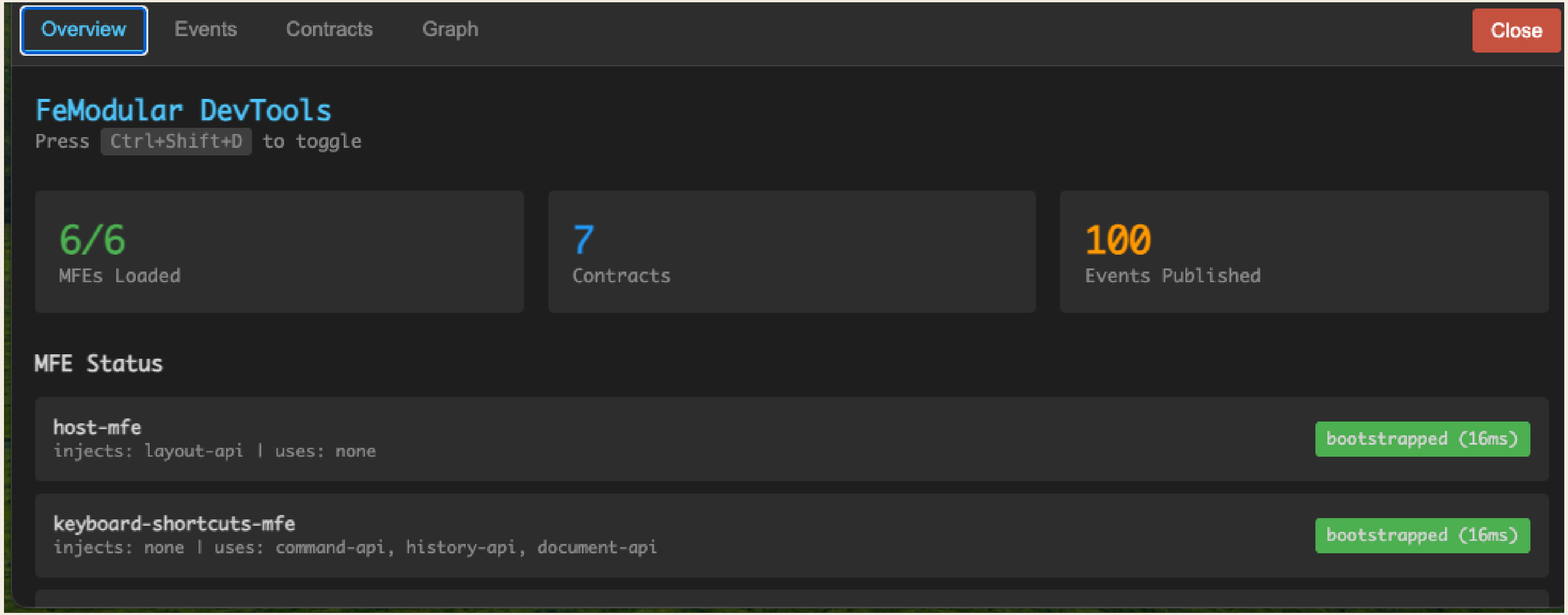
ФАБРИКИ КОНТРАКТІВ

```ts
export class ContractFactory {
  static create<T>(id: string): Contract<T> {
    return { id } as Contract<T>;
  }
}
export interface Contract<T> {
  id: string;
  _apiType?: T;
}
```

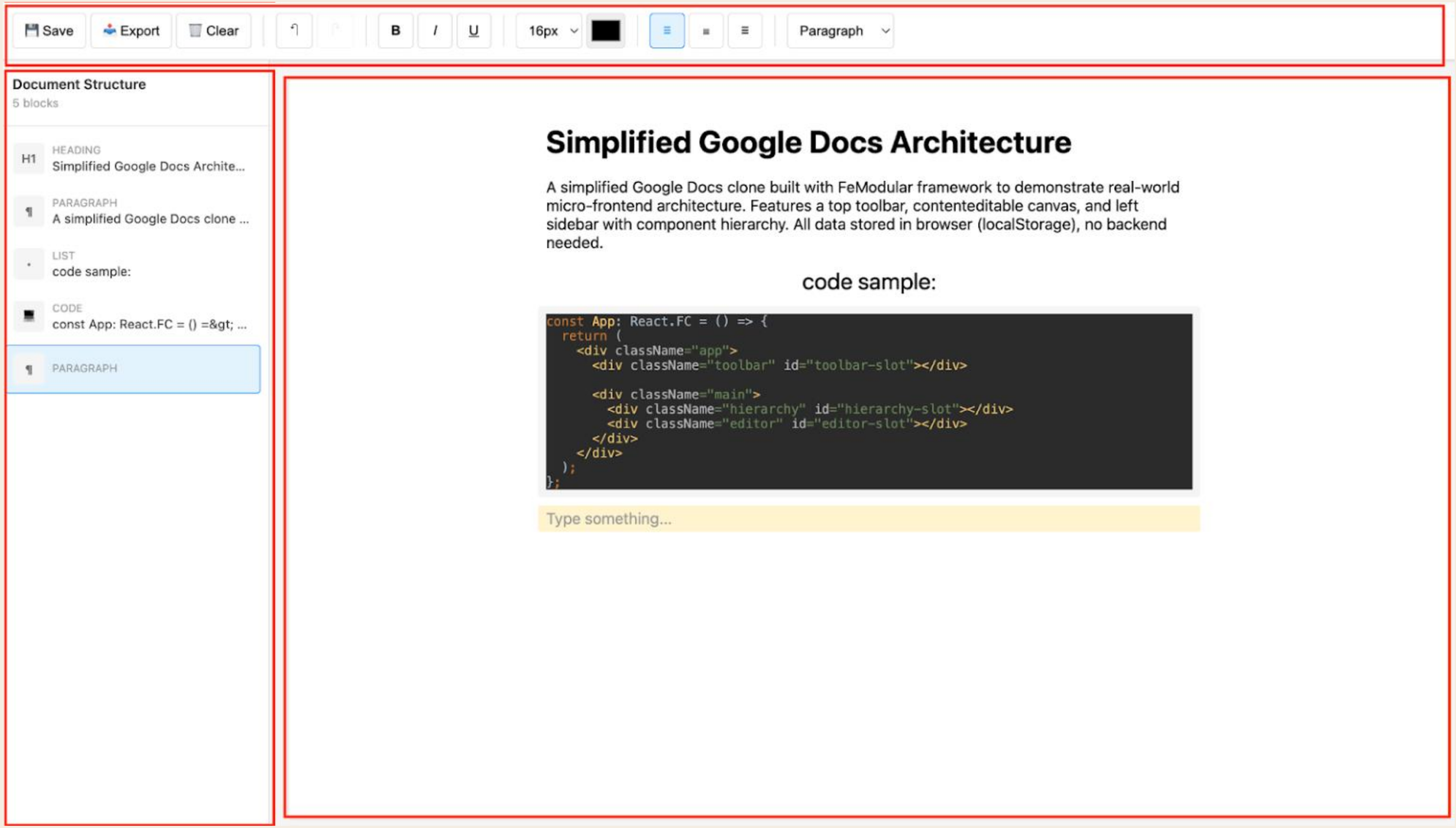
```ts
export class EventFactory {
  static create<T>(type: string): EventContract<T> {
    return { type } as EventContract<T>;
  }
}

export interface EventContract<T> {
  type: string;
  _payloadType?: T;
}
```

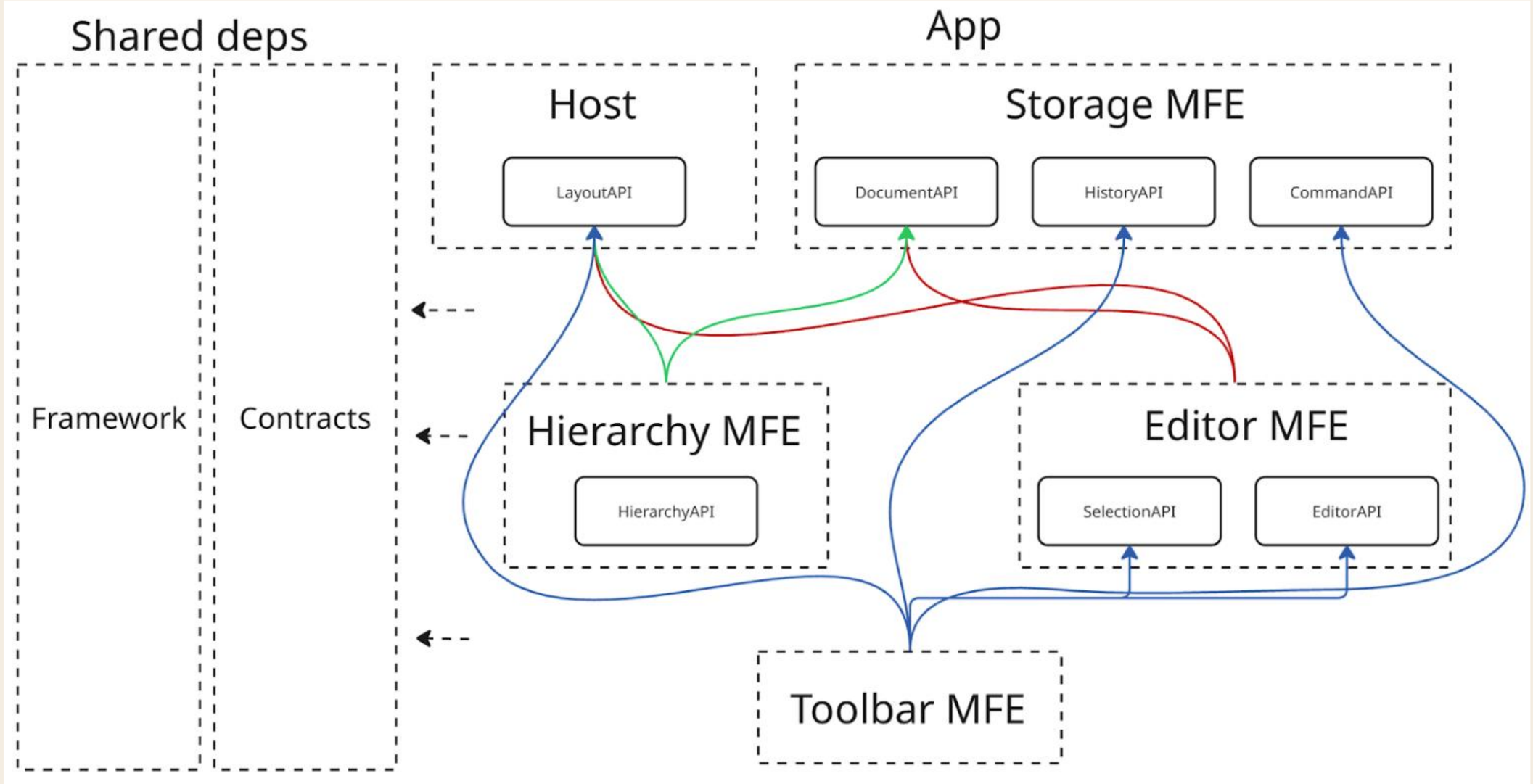
ІНСТРУМЕНТИ РОЗРОБНИКА



ТЕСТОВИЙ ЗАСТОСУНОК



АРХІТЕКТУРА ЗАСТОСУНКУ



ПРИКЛАД ІНТЕГРАЦІЇ

```
export default [  no usages  yuriimysko
    {
        injects: [EditorContract, SelectionContract],
        uses: [LayoutContract, DocumentContract],

        bootstrap: (context: GlobalContext) : Promise<void> | void  => {
            const layoutAPI : LayoutAPI  = context.use(LayoutContract);
            const editorAPI : EditorService  = new EditorService(context);
            const selectionAPI : SelectionService  = new SelectionService();
            context.register(EditorContract, editorAPI);
            context.register(SelectionContract, selectionAPI);

            const app : App<Element>  = createApp(Editor, createEditorProps(context));
            app.mount(layoutAPI.getEditorSlot());
        },
    }
] satisfies MFE[];
```

НАУКОВА НОВИЗНА

1.Здійснено комплексне перенесення всіх шести принципів мікросервісної архітектури на фронтенд з урахуванням специфіки браузерного середовища. Існуючі фреймворки порушують деякі з них.

2. Запропонований метод інтеграції різних UI-фреймворків без спеціалізованих адаптерів через делегування рендерингу самим модулям за допомогою наданого механізму контрактів.

ВИСНОВКИ

- Створено фреймворк, який реалізує всі шість принципів мікросервісів на фронтенді.
- Розроблено систему контрактів, що забезпечує типобезпечну синхронну і асинхронну комунікацію між модулями.
- Забезпечено незалежність від UI технологій та фреймворків.
- Реалізовано підтримку вертикальної та горизонтальної стратегій поділу на модулі.
- Створено інструменти автоматизації конфігурації та налагодження.
- Все це валідовано через реальний демонстраційний застосунок на чотирьох різних технологіях.

ДЯКУЮ ЗА УВАГУ