

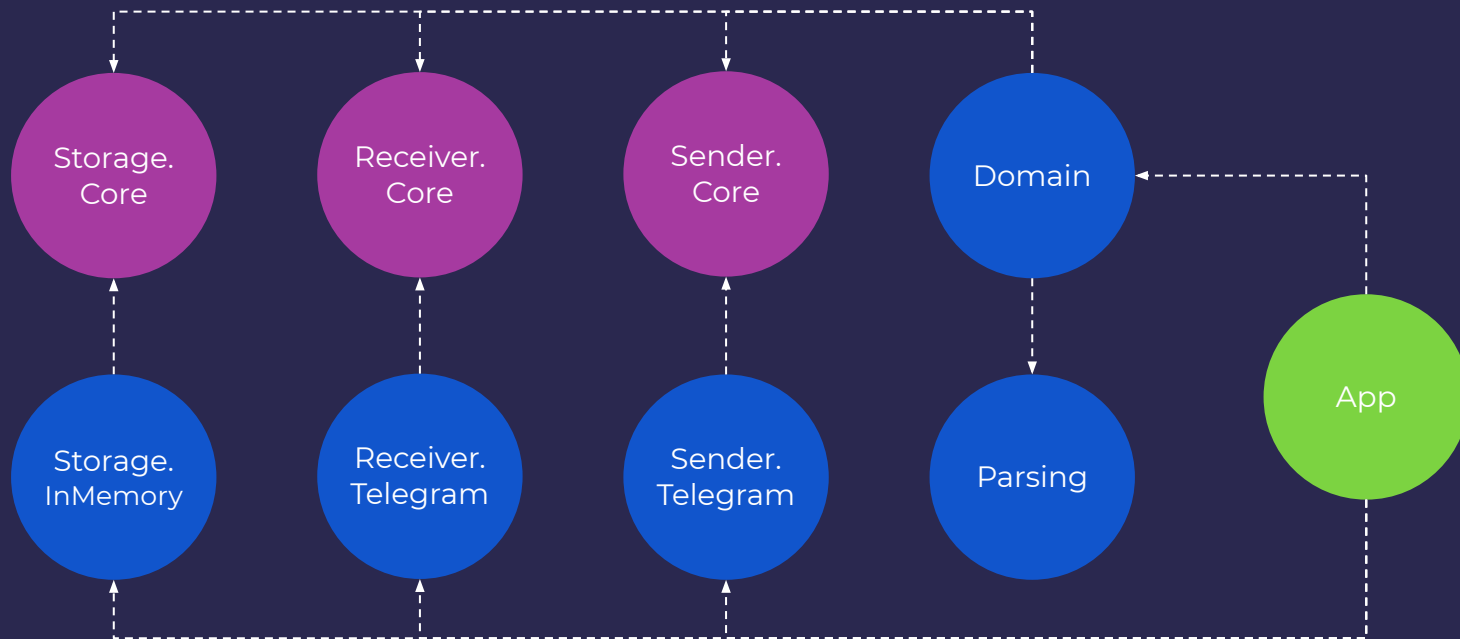


# ASP.NET Core MVC

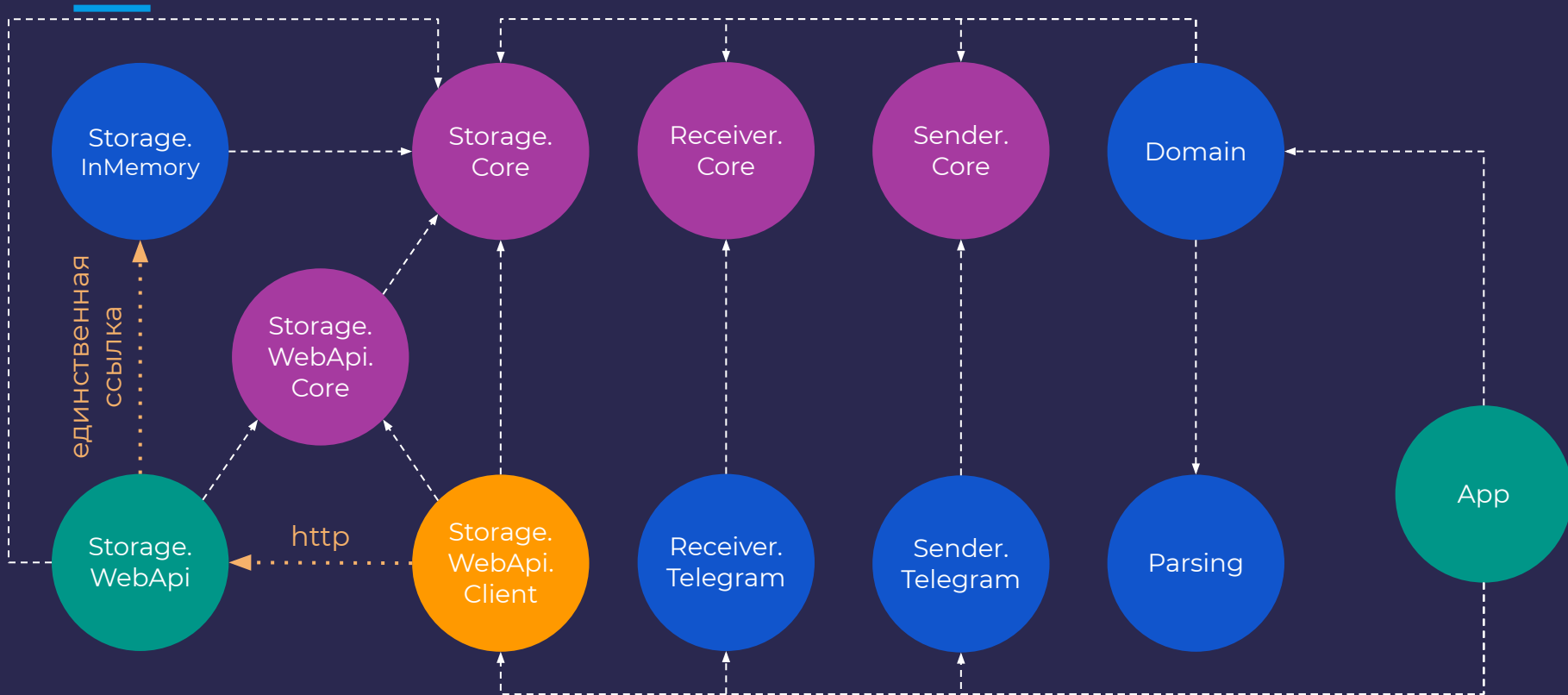
(Организация слоя данных чат-бота как  
отдельного сервиса Web API)

Андрей Голяков

# Исходные взаимосвязи компонентов



# Целевые взаимосвязи компонентов



# Подготовка

---

Делим solution на два (каждый в своей папке):

- Папка **Reminder.App** — для проектов союшена **Reminder.App.sln** основного консольного приложения
  - Удаляем из **Reminder.App.sln** все ссылки на проекты **Reminder.Storage.\***
- Папка **Reminder.Storage** — для проектов союшена Web API, но поскольку у нас там только **Reminder.Storage.\*** проекты, союшн назовём **Reminder.Storage.sln**
  - Здесь будут все наши **Reminder.Storage.\*** проекты



# Вместе пишем много кода...

---

Мы напишем три новых сборки:

- `Reminder.Storage.WebApi` — Web API к нашему хранилищу
- `Reminder.Storage.WebApi.Client` — клиентская библиотека упрощающая работу с нашим Web API
- `Reminder.Storage.WebApi.Core` — сюда в процессе написания перенесётся общий код между первыми двумя сборками



# IReminderStorage

Методы необходимые  
приложению Reminder.App

```
// POST api/reminders
void Add(ReminderItem reminder);

// GET api/reminders?[filter]status=1
List<ReminderItem> Get(ReminderItemStatus status);

// GET api/reminders/{id}
ReminderItem Get(Guid id);

// PATCH api/reminders
// ids = List<id>
// patchDoc = JSON Patch Doc
void UpdateStatus(IEnumerable<Guid> ids, ReminderItemStatus status);

// PATCH api/reminders/{id}
// JSON Patch Doc
void UpdateStatus(Guid id, ReminderItemStatus status);
```



# Домашняя работа

---

Раскомментировать временно закомментированные методы `IReminderStorage` и поддержать их и Web API и в клиентской библиотеке.

Это “д/з без оценки”, скорее для саморазвития, как идея, над чем можно ещё поработать. В качестве примера реализации для самопроверки я приложу выполненную работу к уроку.



# Спасибо за внимание.

