# Основные термины и тезисы

### [Пользователь API](#ПользовательAPI) - это приложение (или набор приложений), использующее выданный ему [ApiKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#ApiKey) для работы с [программным комплексом](#_Программный_комплекс_-).

### [Программный комплекс](#ПрограммныйКомплекс) - это набор подсистем (сервисов и баз данных), обслуживающий несколько [пользователей API](#_Пользователь_API_-). Ничто не мешает развернуть несколько *программных комплексов*, обслуживающих разные пулы [пользователей API](#_Пользователь_API_-). Данные разных [пользователей API](#_Пользователь_API_-) в рамках одного *программного комплекса* изолированы друг от друга, хотя и могут храниться в одной базе данных. Данные разных *программных комплексов* полностью изолированы друг от друга вплоть до того, что хранятся в разных базах данных.

### [Внутренний API](#ВнутреннийAPI) - набор методов [программного модуля](#_Программный_модуль_-) для взаимодействия с ним на уровне сборки, подключённой к проекту. (Фактически это просто программный интерфейс или набор интерфейсов для работы с модулем).

### [Внешний API](#ВнешнийAPI) - набор методов сервисов [распределённой подсистемы](#_Распределённая_подсистема_-) для взаимодействия с ней со стороны удалённого клиента. Доступ к этим методам требует аутентификации и может реализовывать несколько протоколов доступа (WCF, WebApi Json, WebApi Xml, и.т.д).

### [Программный модуль](#ПрограммныйМодуль) - логически обособленный набор программной логики со своим [внутренним API](#_Внутренний_API_-). *Программный модуль* может быть изолированным (не предусматривает управление внешними клиентами) и распределённым. В случае, если модуль распределённый - в нём должен быть описан, также его [внешний API](#_Внешний_API_-), а также разработана клиентская часть модуля (которая будет входить в Navis API SDK), реализующая [внутренний API](#_Внутренний_API_-) модуля путём вызова соответствующий методов [внешнего API](#_Внешний_API_-).

### [Распределённая подсистема](#РаспределённаяПодсистема) - набор [программных модулей](#_Программный_модуль_-), которые хостятся в отдельном приложении. Среди этих [программных модулей](#_Программный_модуль_-) должен быть как минимум один распределённый [модуль](#_Программный_модуль_-). Набор [внешних API](#_Внешний_API_-) распределённых модулей составляет [внешний API](#_Внешний_API_-) *подсистемы*. Каждая такая *подсистема* должна быть приписана к одному или нескольким [пользователям API](#_Пользователь_API_-), которых она может обслуживать. Функциональность *подсистемы* определяется только модулями из которых она состоит. [Программный комплекс](#_Программный_комплекс_-) может иметь несколько абсолютно одинаковых *подсистем* для балансировки нагрузки, однако даже две одинаковые *подсистемы* могут иметь разные базы данных (например две телематические подсистемы имеют разные базы данных и обслуживают разные непересекающиеся пулы [пользователей API](#_Пользователь_API_-)). [Программный комплекс](#_Программный_комплекс_-) должен иметь как минимум одну *подсистему* - [подсистему биллинга](#_Подсистема_биллинга), содержащую в своём составе модули [биллинга](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/BillingAPI.docx) и [сообщений](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/EventsAPI.docx).

### [Подсистема биллинга](#ПодсистемаБиллинга) - единственная обособленная подсистема на весь [программный комплекс](#_Программный_комплекс_-). Помимо [биллинга](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/BillingAPI.docx), координирует взаимодействие между [клиентскими приложениями](#_Клиентское_приложение_-) и остальными [подсистемами](#_Подсистема_биллинга_-), а именно передачу сообщений между [клиентскими приложениями](#_Клиентское_приложение_-), а также координацию (балансировку) доступа к [API подсистем](#_Распределённая_подсистема_-). Данная подсистема должна работать 24/7 и быть устойчива к сбоям, нагрузкам и [DoS-атакам](http://ru.wikipedia.org/wiki/DoS-%D0%B0%D1%82%D0%B0%D0%BA%D0%B0). Подробнее про подсистему [читайте ниже](#_Подсистема_биллинга).

### [Клиентское приложение](#КлиентскоеПриложение) - часть программного комплекса, использующая API модулей подсистем этого комплекса. В самом общем случае - это может быть обычной [распределённой подсистемой](#_Распределённая_подсистема_-), однако мы будем называть *клиентским приложением* изолированную подсистему, т.е. подсистему не имеющую [внешнего API](#_Внешний_API_-). В отличие от обычной [распределённой подсистемы](#_Распределённая_подсистема_-), каждое *клиентское приложение* работает от лица одного конкретного [пользователя API](#_Пользователь_API_-). Может быть несколько приложений, работающих от лица одного конкретного [пользователя API](#_Пользователь_API_-).

Подытаживая вышесказанное - программный комплекс состоит из трёх основных групп:

1. Подсистема биллинга
2. [Распределённые подсистемы](#_Распределённая_подсистема_-) *(опционально)*
3. [Клиентские приложения](#_Клиентское_приложение_-)

# Общая схема программного комплекса

На данной схеме представлена общая схема взаимодействия между различными частями [программного комплекса](#_Программный_комплекс_-). Взаимодействия между [клиентскими приложениями](#_Клиентское_приложение_-) и [подсистемами](#_Распределённая_подсистема_-) осуществляется напрямую. Это касается как вызовов методов API подсистем, так и передачи [сообщений](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/EventsAPI.docx) и работе с [репозиториями](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/ReportsAPI.docx) бизнес-объектов подсистем (для этого подсистемы должны включать в себя [модуль сообщений](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/EventsAPI.docx) и [модуль репозитория](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/ReportsAPI.docx)). Взаимодействия между двумя [клиентскими приложениями](#_Клиентское_приложение_-) может осуществляться только на уровне передачи сообщений и осуществляется через [подсистему биллинга](#h.i2tc17wm64ml).

В настоящее время реализация распределённых подсистем требует наличия рядом с исполнимым файлом-хоста специального файла SubSystem.json с указанием уникального идентификатора подсистемы, а также (опционально) перечня ApiKey’s с которыми может работать данная подсистема.

# Общая схема Пример веб приложения gps мониторинга

# 

Как видно, на рисунке типичный программный комплекс состоит из трёх независимых и распределённых подсистем:

* [Подсистемы биллинга](#_Подсистема_биллинга)
* [Телематической подсистемы](#_Телематическая_подсистема)
* [Клиентского WEB-приложения](#_Клиентское_приложение_(NavisWeb))

# Подсистема биллинга

Подсистема предназначена для регистрации и управления приложениями, пользователями, их тарифными планами и ограничениями. Подсистема имеет внешний **АPI**, однако этот **API** в полном объёме будет использоваться только нами. Конечные пользователи будут ограничены информационными функциями, как-то узнать текущий тарифный план и его детали. Для доступа ко внешнему **API** сторонними пользователями приложений требуется [ApiKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#ApiKey) или [AppKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#AppKey).

Создание нового приложения на платформе **Navis3** должно начинаться с регистрации приложения в этой подсистеме, при которой будет сгенерированы его [ApiKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#ApiKey) и [AppKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#AppKey). [ApiKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#ApiKey) - это конфиденциальная информация, которая однозначно идентифицирует приложения как пользователя **API**. Для того, чтобы она не была скомпрометирована необходимо использовать защищённые протоколы для работы с **API** (например, **HTTPS**). Если требуется доступ к АPI из клиентских приложений - публикуется [AppKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#AppKey) и в дальнейшем используется [AppUser авторизация](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#AppUserАвторизация).

Подсистема имеет собственную базу данных и абсолютно не зависит от других частей программного комплекса.

## Зависимость от подсистем программного комплекса

* нет

## [Внешний API](https://d.docs.live.net/2977baf408e5856c/Navis3/API%20docs/BillingAPI.docx)

# Телематическая подсистема

Телематическая подсистема требует [ApiKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#ApiKey) или [AppKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#AppKey) для работы с её внешним **API**.

Может быть несколько телематических подсистем на одном или разных компьютерах, каждая из которых может обслуживать от одного до нескольких приложений.

При старте телематическая подсистема обращается к подсистеме биллинга, для чтения текущих ограничений для обслуживаемых им приложений. При этом она использует **SSID-аутентификацию** для авторизации.

Это могут быть следующие ограничения:

* Количество УС
* Общее количество хранимых координат
* Время хранения координат
* Бан-лист IP-адресов и идентификаторов УС

## Зависимость от подсистем программного комплекса

* [подсистема биллинга](#_Подсистема_биллинга)

## Внешний API

* Devices - CRUD
* WriteCoordinates
* GetFilters
* ReadCoordinater + Filters
* Subscribe

# Клиентское приложение (NavisWeb)

При создании нового приложения на платформе **Navis3** требуется зарегистрировать его в подсистеме биллинга.

Приложение - это основная часть программного комплекса, то, ради чего всё затевается. Приложений может быть множество и создаваться они могут сторонними разработчиками.

Здесь будет описано конкретное приложение **NavisWeb**, разрабатываемая нами, как одна из возможных подсистем программного комплекса.

Само приложение имеет внешний **API** для управления им извне. Для доступа к внешнему **API** приложения НЕ ТРЕБУЕТСЯ [ApiKey](https://d.docs.live.net/2977baf408e5856c/Navis3/SoftWare%20Architecture/Navis-аутентификация.docx#ApiKey) (ведь это секретная информация). Вместо этого требуется авторизация, реализованная на уровне и средствами Web-приложения. Эта авторизация никак не связана с Navis-авторизацией.

Приложение NavisWeb - это пример приложения, которое может предоставлять пользователям вторичный сервис (**высокоуровневый API**, заточенный под конкретную предметную область и использующий **низкоуровневый API** подсистем программного комплекса).

## Зависимость от подсистем программного комплекса

* [подсистема биллинга](#_Подсистема_биллинга)
* [телематическая подсистема](#_Телематическая_подсистема)