

MOLECULER

MICROSERVICES FRAMEWORK

Привет! Меня зовут Иван Журавлёв!

Более 8 лет занимаюсь архитектурой высоконагруженных и распределённых систем.

Со мной можно обсудить:

- Распределённые транзакции и SAGA pattern;
- Алгоритмы консенсусов;
- Построение современного потокового озера данных;
- Распределённую трассировку;
- Chaos engineering и fault tolerance patterns;



Core maintainer MoleculerJS

За 3 года 5 стартапов из которых посмотреть можно только 3

В феврале 2021 года я присоединился к разработчикам фреймворка

Занимаюсь локализацией документации, русскоязычным комьюнити в телеграмме, багами и сопровождением пользователей



Какая разница между **сервисом** и **микросервисом**?

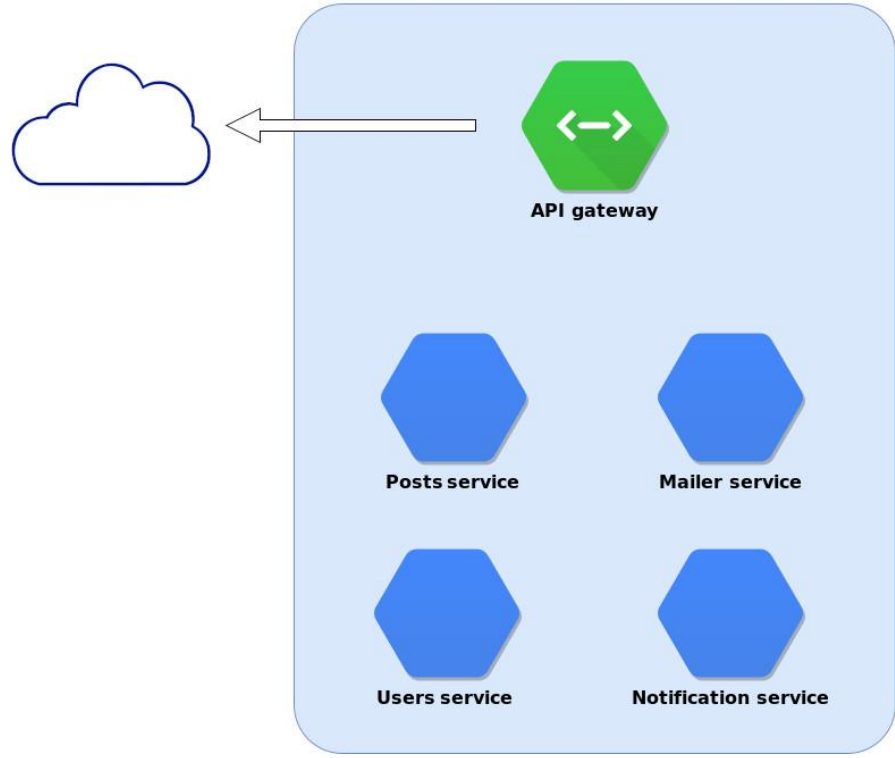
SOA (сервис-ориентированная архитектура)

и

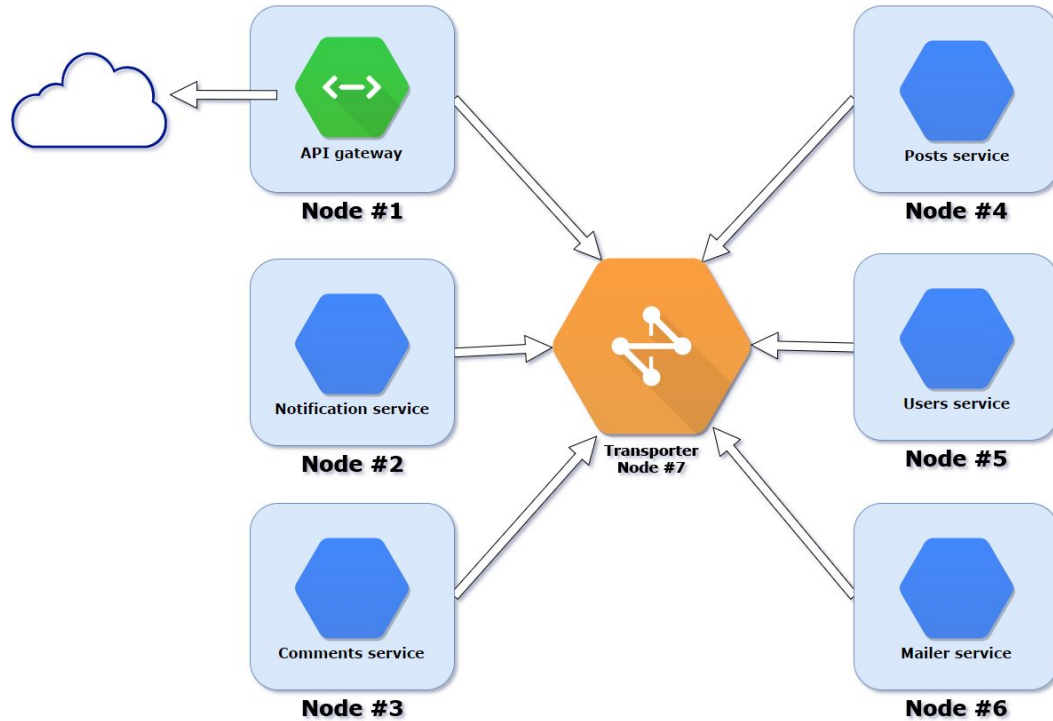
Микросервисная архитектура



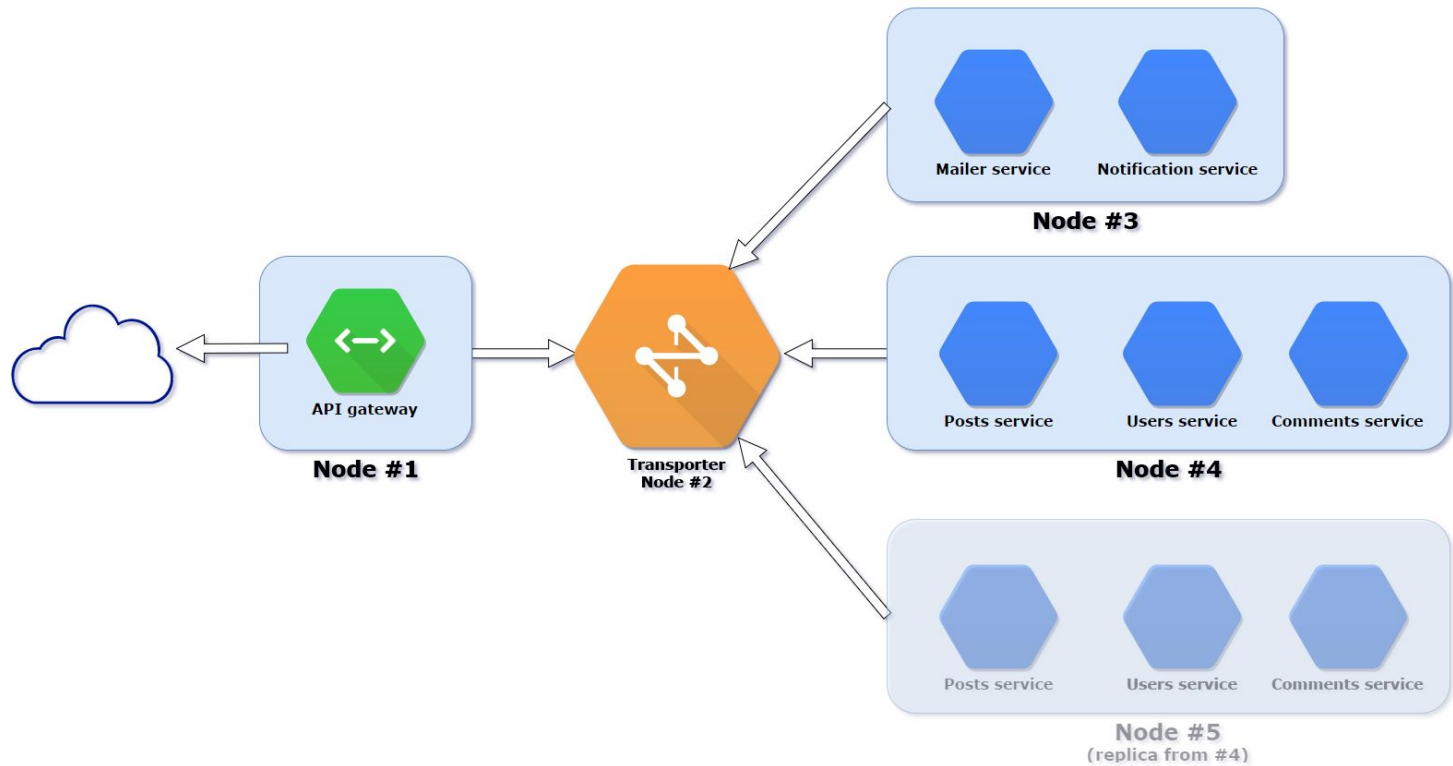
Сервис-ориентированная архитектура (SOA)



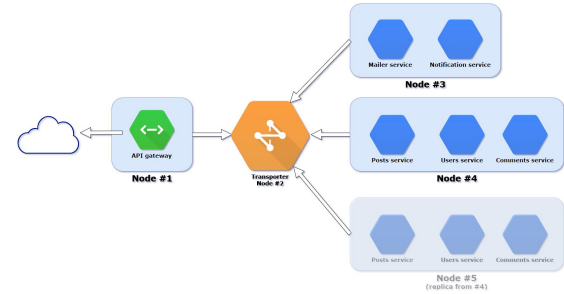
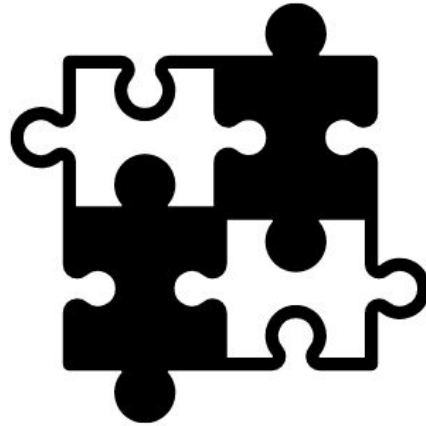
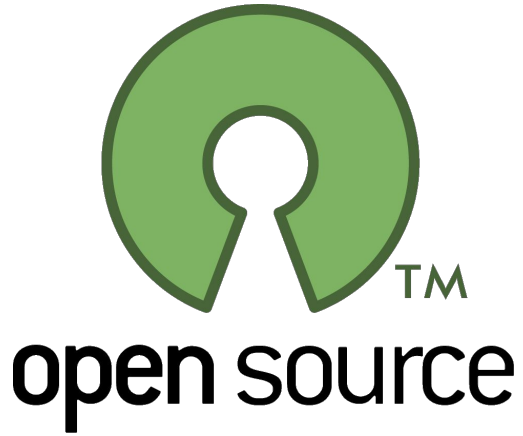
Микросервисная архитектура



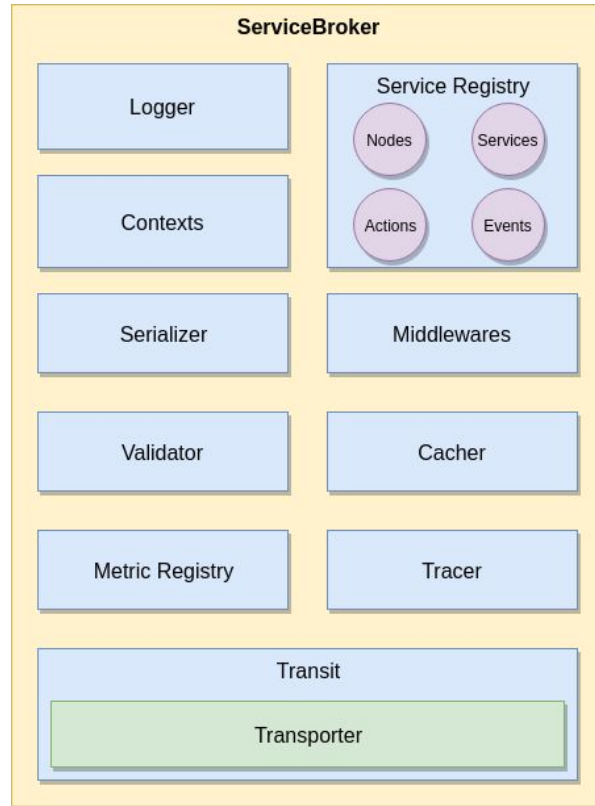
Смешанная/гибридная архитектура



Что такое фреймворк Moleculer?



Основные компоненты



Кэширование

- Собственные
 - Memory (по умолчанию)
 - MemoryLRU
 - Redis (Cluster)
- Кастомная реализация
- Использование распределённых блокировок
- Переопределяется индивидуально в сервисах



Логирование

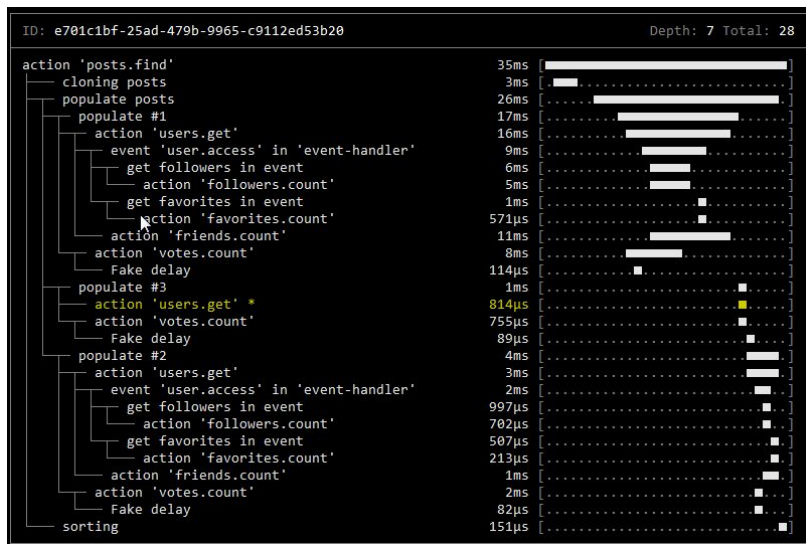
- Собственные
 - Консольный (по умолчанию)
 - Файловый
- Внешние
 - Pino
 - Bunyan
 - Winston
 - debug
 - Log4js
 - Datadog
- Кастомная реализация
- Использование нескольких

```
[2019-09-07T10:22:00.054Z] INFO my-node/BROKER: Moleculer v0.14.0-beta3 is starting...
[2019-09-07T10:22:00.056Z] INFO my-node/BROKER: Namespace: <not defined>
[2019-09-07T10:22:00.057Z] INFO my-node/BROKER: Node ID: my-node
[2019-09-07T10:22:00.058Z] INFO my-node/REGISTRY: Strategy: RoundRobinStrategy
[2019-09-07T10:22:00.063Z] INFO my-node/BROKER: Cacher: MemoryCacher
[2019-09-07T10:22:00.064Z] INFO my-node/BROKER: Serializer: JSONSerializer
[2019-09-07T10:22:00.066Z] INFO my-node/BROKER: Registered 12 internal middleware(s).
[2019-09-07T10:22:00.068Z] INFO my-node/BROKER: Transporter: NatsTransporter
[2019-09-07T10:22:00.072Z] DEBUG my-node/$NODE: Service '$node' is creating...
[2019-09-07T10:22:00.073Z] DEBUG my-node/$NODE: Service '$node' created.
[2019-09-07T10:22:00.075Z] DEBUG my-node/V2.GREETER: Service 'v2.greeter' is creating...
[2019-09-07T10:22:00.075Z] DEBUG my-node/V2.GREETER: Service created!
[2019-09-07T10:22:00.076Z] DEBUG my-node/V2.GREETER: Service 'v2.greeter' created.
[2019-09-07T10:22:00.077Z] DEBUG my-node/TEST: Service 'test' is creating...
[2019-09-07T10:22:00.077Z] DEBUG my-node/TEST: Service created!
[2019-09-07T10:22:00.078Z] DEBUG my-node/TEST: Service 'test' created.
[2019-09-07T10:22:00.079Z] DEBUG my-node/HELLO: Service 'hello' is creating...
[2019-09-07T10:22:00.079Z] DEBUG my-node/HELLO: Service created!
[2019-09-07T10:22:00.080Z] DEBUG my-node/HELLO: Service 'hello' created.
[2019-09-07T10:22:00.081Z] TRACE my-node/MY.CUSTOM.MODULE: Trace test
[2019-09-07T10:22:00.081Z] DEBUG my-node/MY.CUSTOM.MODULE: Debug test
[2019-09-07T10:22:00.082Z] INFO my-node/MY.CUSTOM.MODULE: Info test
[2019-09-07T10:22:00.082Z] WARN my-node/MY.CUSTOM.MODULE: Warn test
[2019-09-07T10:22:00.083Z] ERROR my-node/MY.CUSTOM.MODULE: Error test { MoleculerClientError: Something happened
  at Object.<anonymous> (d:\Work\moleculer\moleculer\dev\loglevel.js:143:30)
  at Module._compile (internal/modules/cjs/loader.js:776:30)
  at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
  at Module.load (internal/modules/cjs/loader.js:653:32)
  at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
  at Function.Module._load (internal/modules/cjs/loader.js:585:3)
  at Module.require (internal/modules/cjs/loader.js:690:17)
  at require (internal/modules/cjs/helpers.js:25:18)
  at Object.<anonymous> (d:\Work\moleculer\moleculer\dev\index.js:6:1)
  at Module._compile (internal/modules/cjs/loader.js:776:30)
  at Object.Module._extensions..js (internal/modules/cjs/loader.js:787:10)
  at Module.load (internal/modules/cjs/loader.js:653:32)
  at tryModuleLoad (internal/modules/cjs/loader.js:593:12)
  at Function.Module._load (internal/modules/cjs/loader.js:585:3)
  at Module.require (internal/modules/cjs/loader.js:690:17)
  at require (internal/modules/cjs/helpers.js:25:18)
}
```



Метрики и трассировка

- Консольная трассировка
- APM Laboratory
- Внешние адаптеры:
 - Datadog
 - Prometheus
 - StatsD
 - NewRelic
 - Jaeger
 - Zipkin
- Кастомная реализация



Чем же он хорош?

- Простой в изучении;
- Быстрый;
- Множество решений из коробки;
- Удобство разработки и отладки;
- Легко масштабируется;
- Подходит для опытных разработчиков;
- Экосистема и комьюнити;



Простой в изучении

Наш первый микросервис

```
const { ServiceBroker } = require("moleculer");  
  
// Create a ServiceBroker  
const broker = new ServiceBroker();
```



Простой в изучении

Наш первый микросервис

```
const { ServiceBroker } = require("moleculer");

// Create a ServiceBroker
const broker = new ServiceBroker();

// Define a service
broker.createService({
  name: "math",
  actions: {
    add(ctx) {
      return Number(ctx.params.a) + Number(ctx.params.b);
    },
  },
});
```



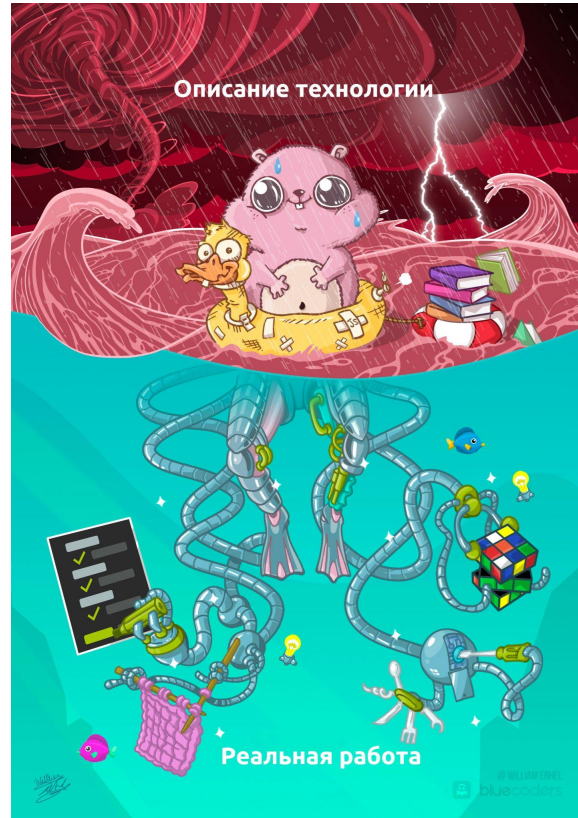
Простой в изучении

Запускаем!

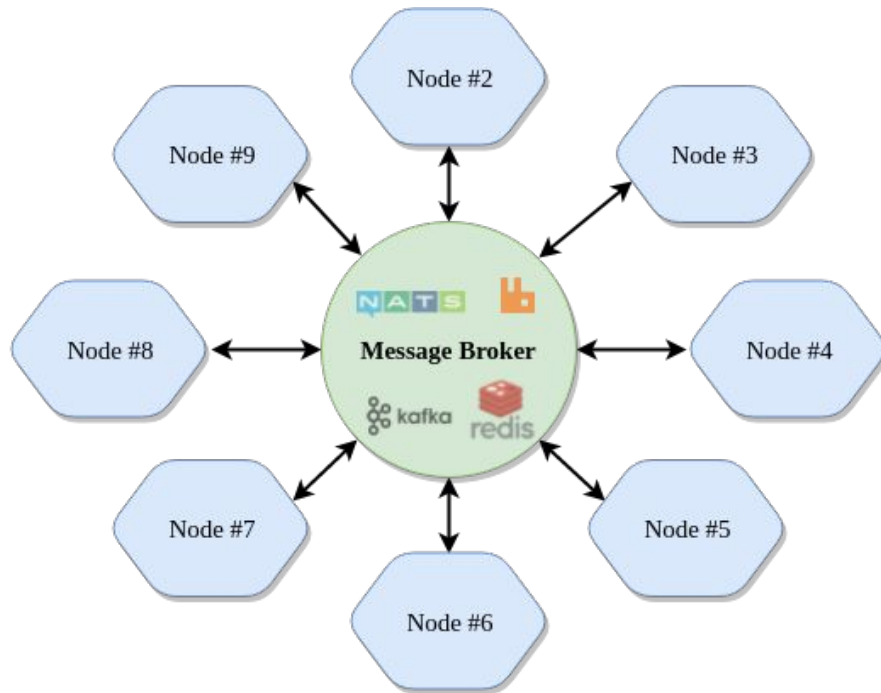
```
// Start the broker
broker
  .start()
  // Call the service
  .then(() => broker.call("math.add", { a: 5, b: 3 }))
  // Print the response
  .then((res) => console.log("5 + 3 =", res))
  .catch((err) => console.error(`Error occurred! ${err.message}`));
```



Дальше начинается hardcore



Транспорт



Транспорт

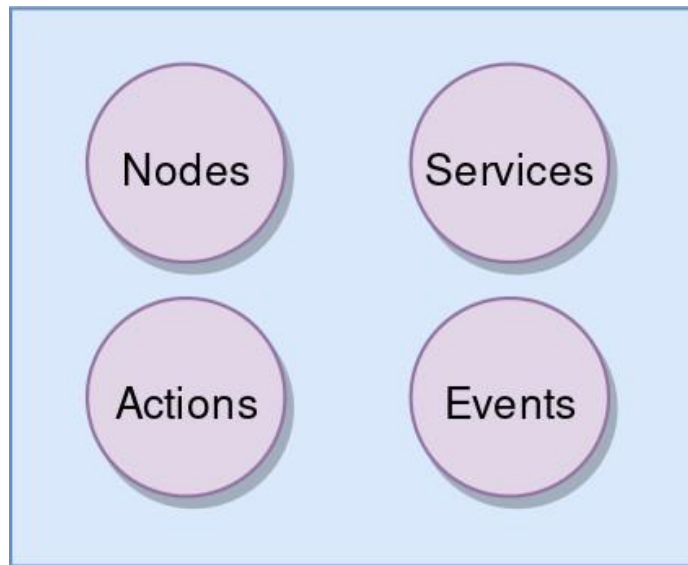
Адаптеры

- Поддерживаемые
 - TCP (GOSSIP)
 - NATS
 - Redis
 - MQTT
 - AMQP (0.9)
 - AMQP 1.0
 - Kafka
 - NATS Streaming (STAN)
- Кастомная реализация



Взаимодействие

Обнаружение и регистрация сервисов



Балансировка

Стратегии балансировки запросов

- Поддерживаемые
 - Round Robin
 - Random
 - Latency
 - По CPU
 - Shard
- Кастомная реализация
- Приоритет балансировки локальный/удалённый



Сериализация

На примере JSON.stringify

```
JSON.stringify({  
  type: "event",  
  seq: 25,  
});  
// {"type":"event","seq":25}
```



Сериализация

Адаптеры

- Поддерживаемые
 - JSON
 - MsgPack
 - Notepack
 - Avro
 - ProtoBuf
 - Thrift
 - CBOR
- Кастомная реализация



Сериализация

Почему не JSON? Пример MsgPack

	JSON	MessagePack
null	null 4 bytes	c0 1 byte
Integer	10 2 bytes	0a 1 byte
Array	[20] 4 bytes	91 14 2 bytes
String	"30" 4 bytes	a2 '3' 3 bytes
Map	{"40":null} 11 bytes	81 a1 '4' 5 bytes



Сериализация

Не только размер, но и скорость

```
JSON length: 169
Avro length: 88
MsgPack length: 126
ProtoBuf length: 90
Thrift length: 134
Notepack length: 126
CBOR length: 133
Suite: Serialize packet with 10bytes
✓ JSON          335 506 rps
✓ Avro          144 780 rps
✓ MsgPack       47 376 rps
✓ ProtoBuf     214 795 rps
✓ Thrift        53 697 rps
✓ Notepack     267 995 rps
✓ CBOR         570 766 rps

JSON (#)      0%      (335 506 rps)  (avg: 2µs)
Avro          -56,85%  (144 780 rps)  (avg: 6µs)
MsgPack       -85,88%  (47 376 rps)   (avg: 21µs)
ProtoBuf     -35,98%  (214 795 rps)  (avg: 4µs)
Thrift        -84%    (53 697 rps)   (avg: 18µs)
Notepack     -20,12%  (267 995 rps)  (avg: 3µs)
CBOR         +70,12%  (570 766 rps)  (avg: 1µs)
```



Контексты

Передача мета-данных между сервисами

- Формируется при каждом вызове сервиса или события
- Содержит пользовательские параметры и мета-данные
- Позволяет передавать данные от сервиса к сервису дополняя их
- Предоставляет доступ к родительскому контексту
- Хранит информацию для трассировки и метрик



Отказоустойчивость

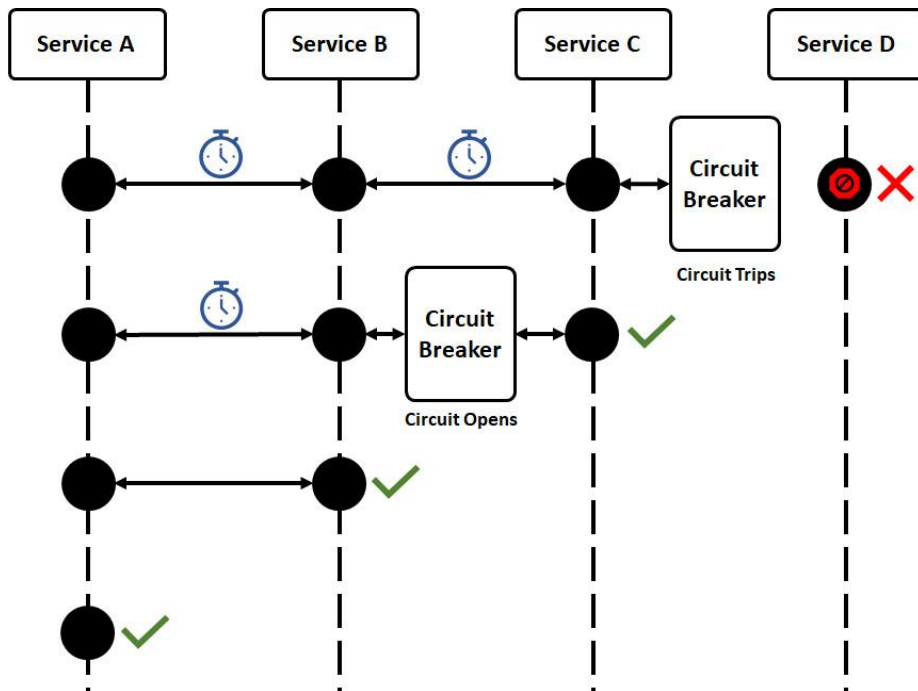
Шаблоны обеспечения отказоустойчивости в микросервисах

- throttle
- debounce
- retry
- fallback
- (distributed) timeout
- bulkhead
- circuit breaker
- call depth limit



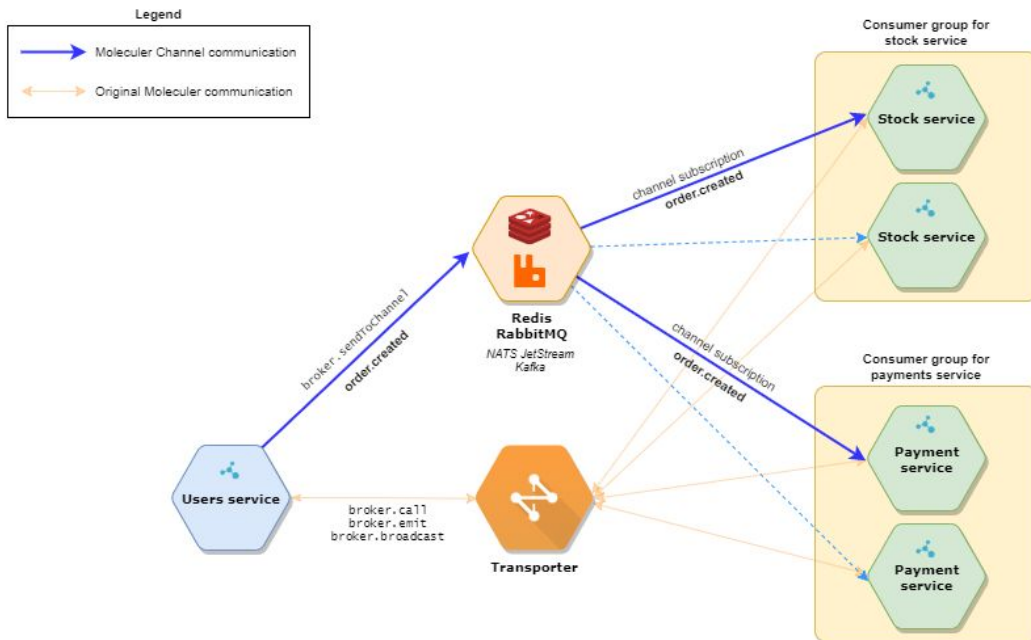
Отказоустойчивость

Circuit Breaker



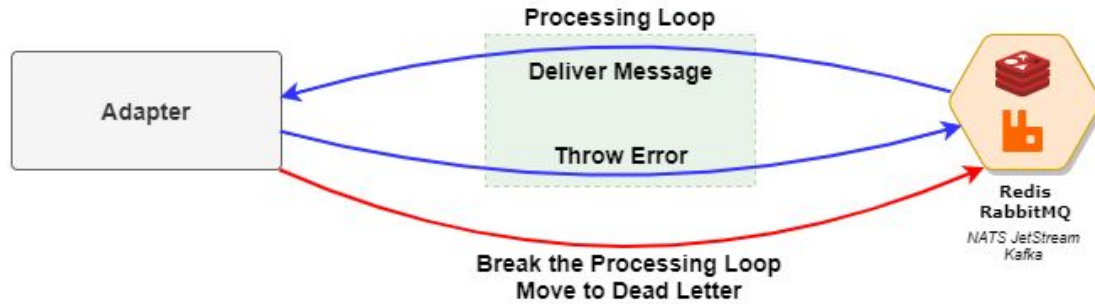
Moleculer Channels

Гарантия доставки событий



Molecular Channels

Очередь недоставленных сообщений (DLQ)



Базы данных

Moleculer-db и Database

Паттерны:

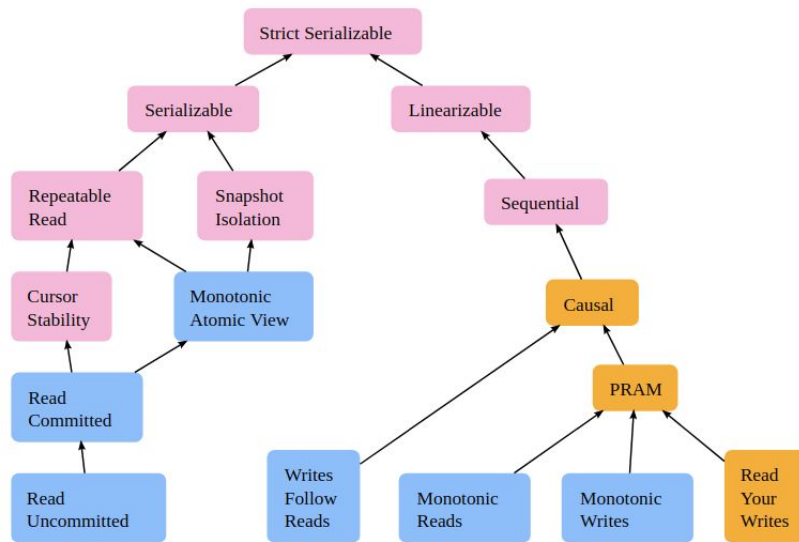
1. База данных на сервис
2. Таблица/коллекция на сервис
3. Общая база данных



Базы данных

Архитектурные проблемы

- Большое количество подключений
- Распределённые транзакции
- SAGA (хореография и оркестрация)
- Chaos engineering и jepsen тест



APM Laboratory

Инструмент для разработчиков

- Карта сети и сервисов
- Метрики по каждому сервису и брокеру
- Трассировка всех внешних и внутренних вызовов
- Логи



Overview

Nodes

Services

Actions

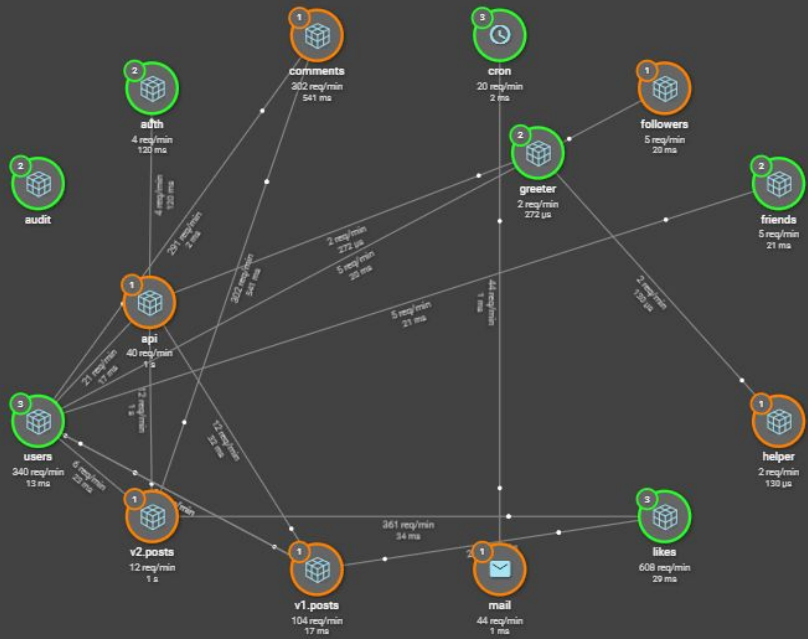
Events

Metrics

Tracing

Logs

Settings



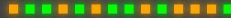
Overview

- Nodes
- Services
- Actions
- Events
- Metrics
- Tracing
- Logs
- Settings

Nodes



Services



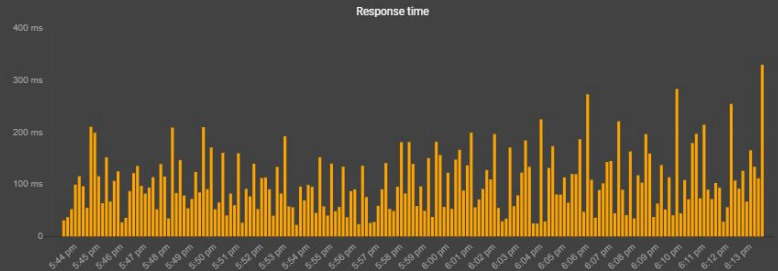
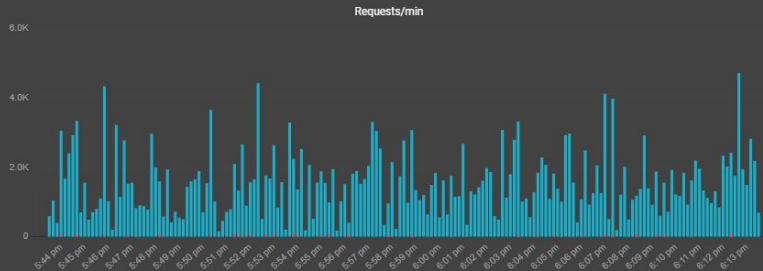
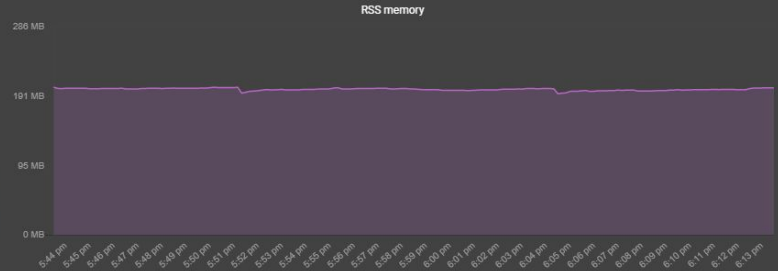
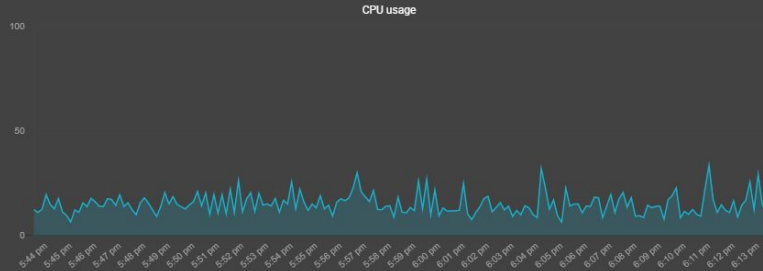
Actions



Events



Metrics



Overview

Nodes

Services

Actions

Events

Metrics

Tracing

Logs

Settings

[← BACK TO TRACE LIST](#)

Trace: GET /api/users/find?limit=5&populate=friends,followers,postCount

de7b0719-6167-4355-8d61-c49d695321f6

SAMPLED AT

6:13:46 PM

a few seconds ago

DURATION

33 ms

33.602 ms

SPANS

17

DEPTH

3



Overview

Nodes

Services

Actions

Events

Metrics

Tracing

Logs

Settings

```
2020-08-08T16:13:40.219Z INFO MAIN-NODE-API <= 200 POST /api/auth/logout?username=icebob [+107.633 ms]
2020-08-08T16:13:40.219Z INFO MAIN-NODE-CLIENT [Virtual client #1] OK! Path: /auth/logout, Time: 180ms
2020-08-08T16:13:40.553Z INFO MAIN-NODE-API => GET /v2/posts/list?page=2&populate=author,likers,comments
2020-08-08T16:13:40.557Z INFO MAIN-NODE-API Call 'v2.posts.list' action
2020-08-08T16:13:40.574Z WARN MAIN-NODE-BROKER Retry to call 'comments.findByPost' action after 500 ms... { requestID: '21656ae1-faad-4967-8944-4182e1874115', attempts: 1 }
2020-08-08T16:13:40.582Z WARN MAIN-NODE-BROKER Retry to call 'comments.findByPost' action after 500 ms... { requestID: '21656ae1-faad-4967-8944-4182e1874115', attempts: 1 }
2020-08-08T16:13:40.919Z INFO MAIN-NODE-API => GET /users/find?limit=10
2020-08-08T16:13:40.923Z INFO MAIN-NODE-API Call 'users.find' action
2020-08-08T16:13:40.927Z INFO MAIN-NODE-API <= 200 GET /api/users/find?limit=10 [+7.825 ms]
2020-08-08T16:13:40.931Z INFO MAIN-NODE-CLIENT [Virtual client #1] OK! Path: /users/find, Time: 14ms
2020-08-08T16:13:41.089Z WARN MAIN-NODE-BROKER Retry to call 'comments.findByPost' action after 1000 ms... { requestID: '21656ae1-faad-4967-8944-4182e1874115', attempts: 2 }
2020-08-08T16:13:42.106Z ERROR MAIN-NODE-API Request error! MoleculerRetryableError: Something happened
MoleculerRetryableError: Something happened
  at Service_handler (D:\Work\moleculer\moleculer-lab\service\example\services\comments.service.js:63:12)
  at D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\utils.js:182:22
  at validateContextParams (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\validators\base.js:72:15)
  at ServiceBroker_timeoutMiddleware (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\timout.js:35:14)
  at ServiceBroker_retryMiddleware (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\retry.js:25:12)
  at ServiceBroker_fallbackMiddleware (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\fallback.js:29:11)
  at ServiceBroker_errorHandlerMiddleware (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\error-handler.js:14:18)
  at ServiceBroker_tracingLocalActionMiddleware (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\tracing.js:96:12)
  at Object.metricsMiddleware [as handler] (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\metrics.js:27:11)
  at ServiceBroker_call (D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\service-broker.js:1084:31)
  at D:\Work\moleculer\moleculer-lab\service\node_modules\moleculer\src\middlewares\retry.js:61:24
  at async D:\Work\moleculer\moleculer-lab\service\example\services\posts.v2.service.js:19:69
  at async Promise.all (index 1)
Data:
2020-08-08T16:13:42.117Z INFO MAIN-NODE-API <= 500 GET /api/v2/posts/list?page=2&populate=author,likers,comments [+1.558 s]
2020-08-08T16:13:42.128Z WARN MAIN-NODE-CLIENT [Virtual client #2] Request error { name: 'MoleculerRetryableError', message: 'Something happened', code: 500, type: 'SOMETHING_HAPPENED' }
2020-08-08T16:13:42.699Z INFO MAIN-NODE-API => POST /auth/logout?username=icebob
2020-08-08T16:13:42.705Z INFO MAIN-NODE-API Call 'auth.logout' action
2020-08-08T16:13:42.738Z INFO MAIN-NODE-API <= 200 POST /api/auth/logout?username=icebob [+36.430 ms]
2020-08-08T16:13:42.746Z INFO MAIN-NODE-CLIENT [Virtual client #2] OK! Path: /auth/logout, Time: 51ms
2020-08-08T16:13:42.749Z INFO MAIN-NODE-API => GET /greeter/welcome?name=Moleculer
2020-08-08T16:13:42.751Z INFO MAIN-NODE-API Call 'greeter.welcome' action
2020-08-08T16:13:42.754Z INFO MAIN-NODE-API <= 200 GET /api/greeter/welcome?name=Moleculer [+4.391 ms]
2020-08-08T16:13:42.757Z INFO MAIN-NODE-CLIENT [Virtual client #1] OK! Path: /greeter/welcome, Time: 15ms
2020-08-08T16:13:43.950Z INFO MAIN-NODE-API => GET /users/list?page=3&populate=friends,followers
2020-08-08T16:13:43.955Z INFO MAIN-NODE-API Call 'users.list' action
2020-08-08T16:13:43.981Z INFO MAIN-NODE-API <= 200 GET /api/users/list?page=3&populate=friends,followers [+32.007 ms]
2020-08-08T16:13:43.985Z INFO MAIN-NODE-CLIENT [Virtual client #2] OK! Path: /users/list, Time: 39ms
2020-08-08T16:13:44.869Z INFO MAIN-NODE-API => GET /users/list?page=2&populate=friends,followers,postCount
2020-08-08T16:13:44.873Z INFO MAIN-NODE-API Call 'users.list' action
2020-08-08T16:13:44.915Z INFO MAIN-NODE-API <= 200 GET /api/users/list?page=2&populate=friends,followers,postCount [+35.653 ms]
2020-08-08T16:13:44.920Z INFO MAIN-NODE-CLIENT [Virtual client #1] OK! Path: /users/list, Time: 55ms
2020-08-08T16:13:46.553Z INFO MAIN-NODE-API => GET /users/find?limit=5&populate=friends,followers,postCount
2020-08-08T16:13:46.556Z INFO MAIN-NODE-API Call 'users.find' action
2020-08-08T16:13:46.584Z INFO MAIN-NODE-API <= 200 GET /api/users/find?limit=5&populate=friends,followers,postCount [+28.211 ms]
2020-08-08T16:13:46.587Z INFO MAIN-NODE-CLIENT [Virtual client #2] OK! Path: /users/find, Time: 36ms
2020-08-08T16:13:49.730Z INFO MAIN-NODE-API => GET /v1/posts/list?page=1&populate=author
2020-08-08T16:13:49.733Z INFO MAIN-NODE-API Call 'v1.posts.list' action
```

Search...



По-настоящему быстрый!

Множество бенчмарков по каждому модулю



Экосистема и комьюнити

Экосистема и комьюнити



Polyglot Implementations

Java

- [moleculer-java](#) - Java implementation of the Moleculer microservices framework.

Python

- [moleculer-python](#) - Python implementation of the Moleculer microservices framework.
- [moleculer-client](#) - Simple Client to communicate with Moleculer services using NATS.

.NET Core

- [moleculer-net](#) - NET Core 2.1 implementation of the Moleculer microservices framework.

Go

- [moleculer-go](#) - Go implementation of the Moleculer microservices framework.

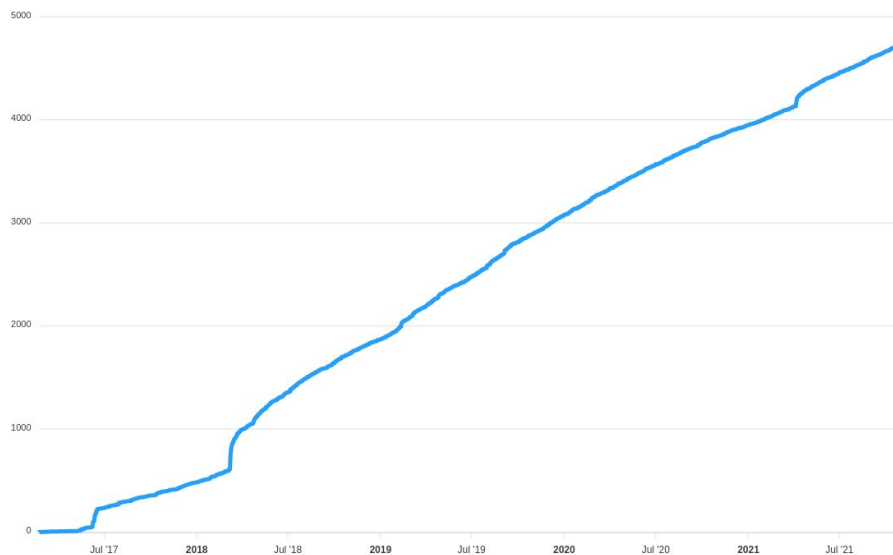
Ruby

- [moleculer-ruby](#) - Ruby implementation of the Moleculer framework.



Экосистема и комьюнити

Линейный рост звёзд на GitHub

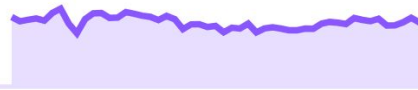


Экосистема и комьюнити

Скачивания на npm

↓ Weekly Downloads

27 383



Version

0.14.18

License

MIT

Unpacked Size

1.16 MB

Total Files

157



Экосистема и комьюнити

Множество проектов в production




Экосистема и комьюнити

Известные в России



Экосистема и комьюнити

И пишут благодарности, например -  **ghost**



pysysops commented on 30 Mar 2019

Contributor



Hey @icebob and @AndreMaz !! Awesome work, there's so much to learn! We're slowly refactoring and improving our hosting orchestration and billing platforms with the help of [moleculer.js](#) over at [Ghost!](#)

Thanks from the team at: <https://github.com/TryGhost> 🤖



Спасибо за внимание

Присоединяйся прямо сейчас



Telegram

Вопросы?



Discord



Контакты

<https://moleculer.services>

Telegram: @ru31337

Email: ru31337@gmail.com

Github: intech

