

Software Engineering Conference Russia

14-15 ноября, 2019. Санкт-Петербург

О разработке прототипа виртуального ассистента для граждан умного города

Евгений Липкин

Университет ИТМО

Кому будет интересен доклад?

1. Разработчикам концепции «Умный город»
2. Разработчикам виртуальных ассистентов
3. Ученым
4. Урбанистам
5. Аналитикам данных
6. Всем, всем, всем, кому интересна концепция «Умный город»

«Умный город» — концепция интеграции **информационно-коммуникационных технологий** и **интернета вещей** для взаимодействия с городской инфраструктурой



О нас

Мы - научный коллектив при университете ИТМО, который решает теоретические (пока что) задачи по реализации концепции «Умный город»

На текущий момент:

1. Два года научных исследований
2. Более 10 научных публикаций
3. Ищем прикладные задачи

Наша задача

Спроектировать систему взаимодействия граждан и сервисов умного города и разработать прототип системы

Типы взаимодействия граждан и сервисов умного города

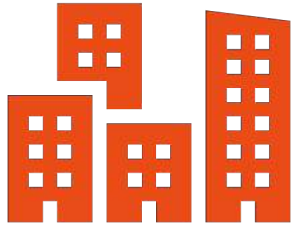
- Предоставление информации
- Поддержка принятия решений

Примеры сервисов умного города

№	Название сервиса	Описание	Входные данные	Выходные данные
7	Сервис велопарковок города	Позволяет получать информацию о велопарковках в городе	Ареал поиска/текущее местоположение пользователя	Карта велопарковок
8	Сервис проката велосипедов	Позволяет узнать места проката велосипедов	Ареал поиска/текущее местоположение пользователя	Карта пунктов проката велосипедов и информация по ним
9	Сервис поиска маршрутов для велосипедистов в городе	Позволяет искать и строить маршруты для велосипедов	Место отправления, место назначения	Маршрут для велосипедиста на карте
10	Сервис проката автомобилей	Позволяет узнать места проката автомобилей	Ареал поиска/текущее местоположение пользователя	Карта пунктов проката автомобилей и информация по ним
11	Сервис проката самокатов	Позволяет узнать места проката самокатов	Ареал поиска/текущее местоположение пользователя	Карта пунктов проката самокатов и информация по ним
12	Сервис поиска попутчиков в городе	Позволяет найти попутчиков	Ареал поиска/текущее местоположение пользователя, место отправления, место назначения	Информация о возможных попутчиках
13	Сервис цен на топливо	Позволяет узнать текущую информацию о ценах на топливо	Текущая дата, текущее время	Информация о текущих ценах на топливо
14	Сервис ближайших станций метро	Позволяет получить информацию о ближайших станциях метро	Текущее местоположение пользователя, текущее время, текущая дата	Карта ближайших станций метро



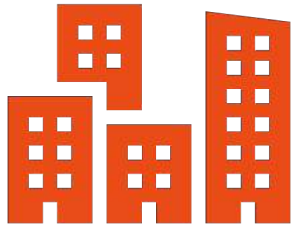
Сервисы умного города



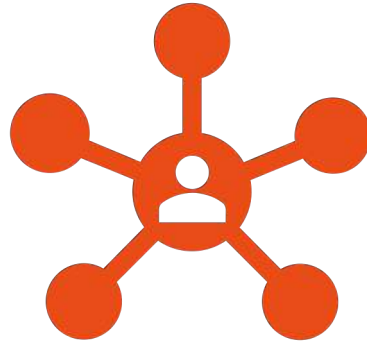
**Сервисы
умного города**



**Запросы
пользователей**



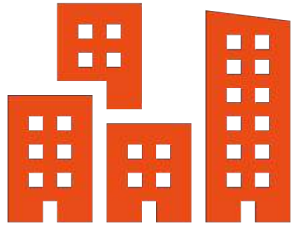
**Сервисы
умного города**



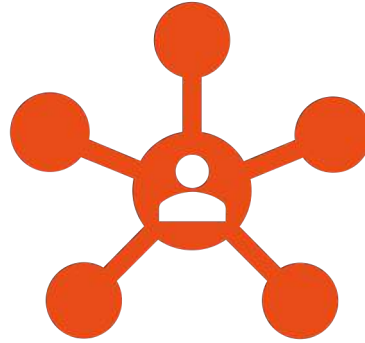
Связывание



**Запросы
пользователей**



**Сервисы
умного города**



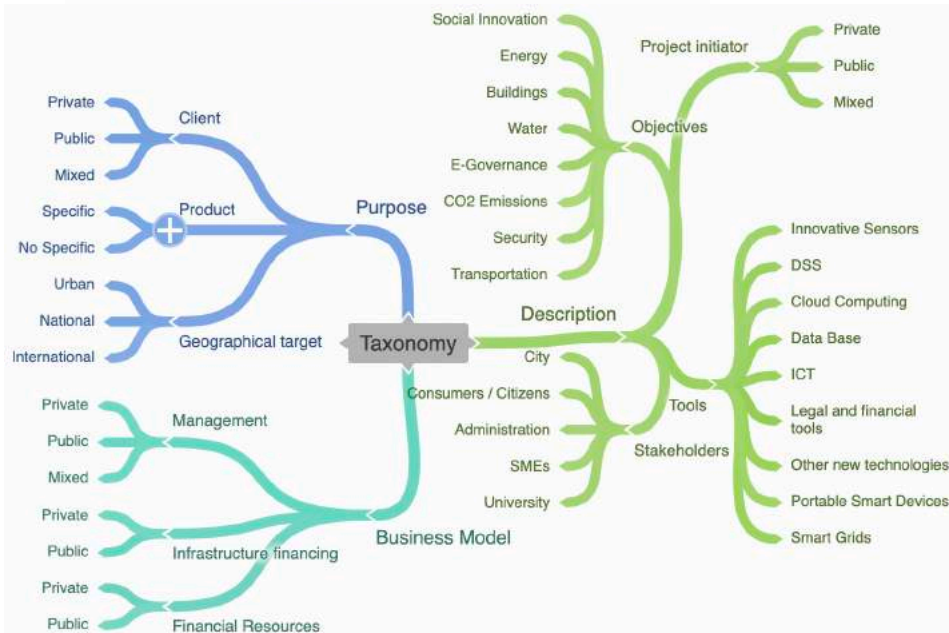
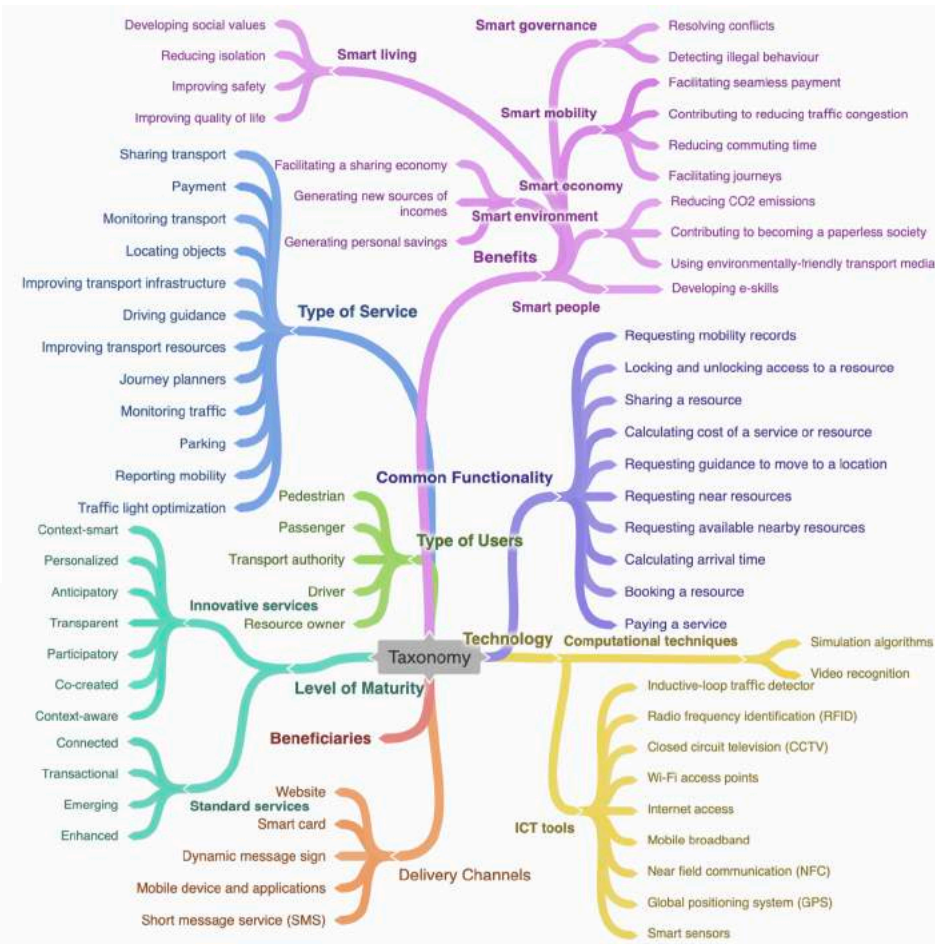
Связывание



**Запросы
пользователей**

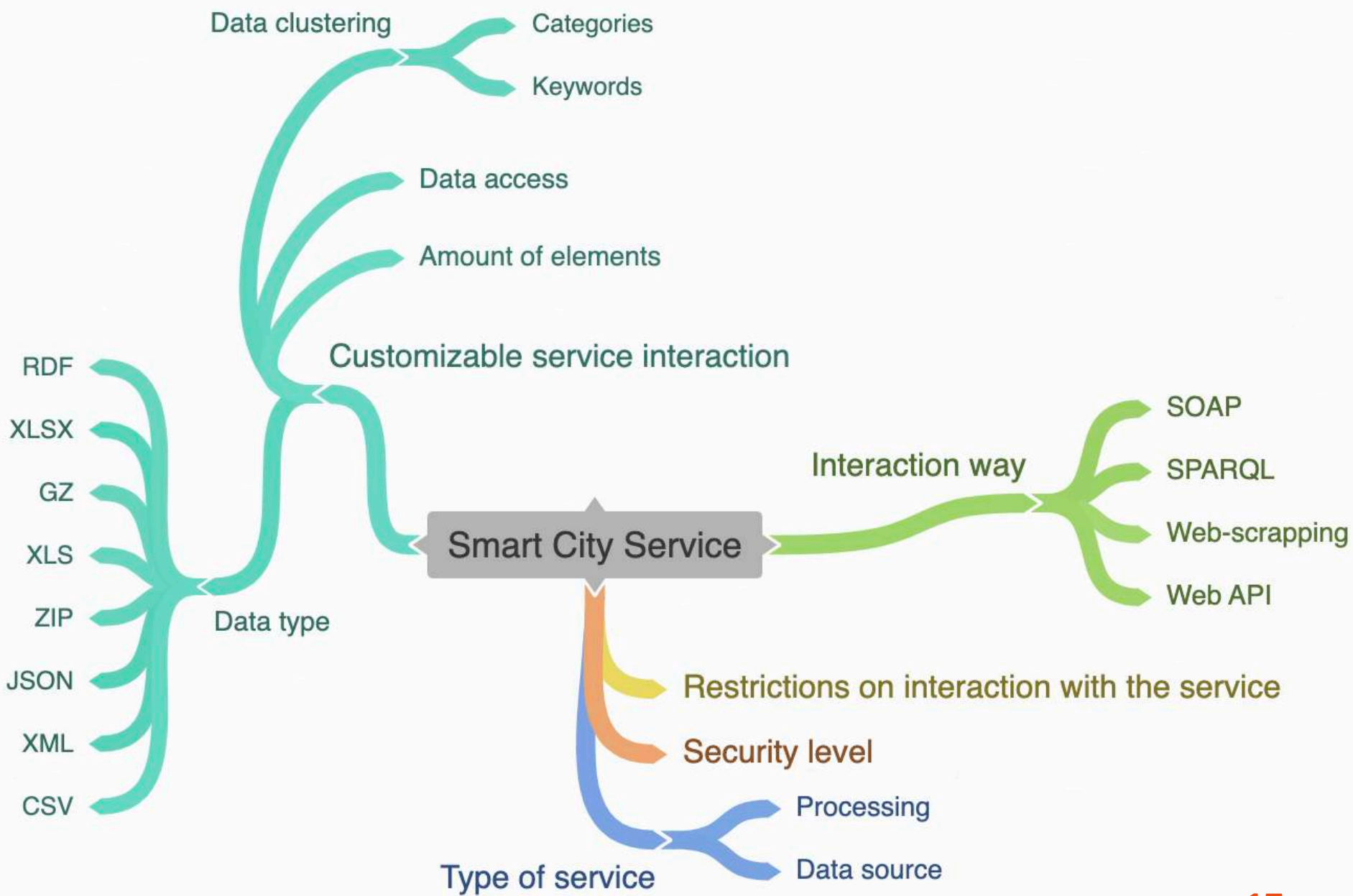


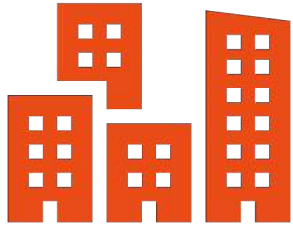
Примеры моделей сервисов умного города



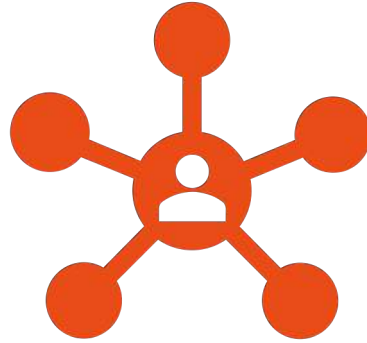
А может лучше разработать
модель абстрактного сервиса умного города и
методы семантического описания сервисов?

Наша модель сервисов умного города





**Сервисы
умного города**



Связывание



**Запросы
пользователей**



Примеры моделей потребностей

Модели потребностей

1. Пирамида Маслоу
2. ERG модель
3. Контекстная таксономия
4. Запросы Yandex



Модели потребностей

1. Пирамида Маслоу
2. **ERG модель**
3. Контекстная таксономия
4. Запросы Yandex



Модели потребностей

1. Пирамида Маслоу
2. ERG модель
3. Контекстная таксономия
4. Запросы Yandex

Unique ID	Parent	Name	Tier 1	Tier 2	Tier 3	Tier 4
54	53	Business Accounting & Finance	Business and Finance	Business	Business Accounting & Finance	
55	53	Human Resources	Business and Finance	Business	Human Resources	
56	53	Large Business	Business and Finance	Business	Large Business	
57	53	Logistics	Business and Finance	Business	Logistics	
58	53	Marketing and Advertising	Business and Finance	Business	Marketing and Advertising	
59	53	Sales	Business and Finance	Business	Sales	
60	53	Small and Medium-sized Business	Business and Finance	Business	Small and Medium-sized Business	
61	53	Startups	Business and Finance	Business	Startups	
62	53	Business Administration	Business and Finance	Business	Business Administration	
63	53	Business Banking & Finance	Business and Finance	Business	Business Banking & Finance	
64	63	Angel Investment	Business and Finance	Business	Business Banking & Finance	Angel Investment
65	63	Bankruptcy	Business and Finance	Business	Business Banking & Finance	Bankruptcy
66	63	Business Loans	Business and Finance	Business	Business Banking & Finance	Business Loans
67	63	Debt Factoring & Invoice Discounting	Business and Finance	Business	Business Banking & Finance	Debt Factoring & Invoice Discounting
68	63	Mergers and Acquisitions	Business and Finance	Business	Business Banking & Finance	Mergers and Acquisitions
69	63	Private Equity	Business and Finance	Business	Business Banking & Finance	Private Equity
70	63	Sale & Lease Back	Business and Finance	Business	Business Banking & Finance	Sale & Lease Back
71	63	Venture Capital	Business and Finance	Business	Business Banking & Finance	Venture Capital
72	53	Business I.T.	Business and Finance	Business	Business I.T.	
73	53	Business Operations	Business and Finance	Business	Business Operations	
74	53	Consumer Issues	Business and Finance	Business	Consumer Issues	
75	74	Recalls	Business and Finance	Business	Consumer Issues	Recalls
76	53	Executive Leadership & Management	Business and Finance	Business	Executive Leadership & Management	
77	53	Government Business	Business and Finance	Business	Government Business	
78	53	Green Solutions	Business and Finance	Business	Green Solutions	

Модели потребностей

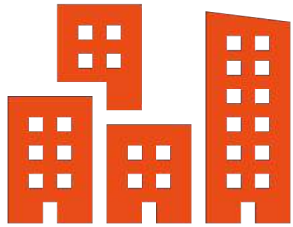
1. Пирамида Маслоу
2. ERG модель
3. Контекстная таксономия
4. Запросы Yandex

что это
что такое
что нужно
что можно
что лучше
что делать
сколько стоит
сколько лет
сколько
своими руками
самый
самостоятельно
почему
откуда
куда
кто это
кто такой
кто
когда
какой
как узнать
как сделать
как правильно
как получить
как найти
как можно
как
зачем
для чего
где находится
где
в домашних условиях

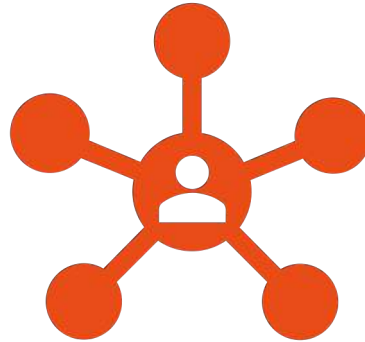
Наша модель пользовательских запросов

Пример исходных данных	Пример запроса пользователя	Пример рекомендации
пользователь собирается куда-то поехать	Что надеть?	погода, одежда, зонт, ветер
	Как доехать?	транспорт, такси, каршэринг
пользователь каждый будний день выходит в 8:00 на работу	Что надеть?	погода, одежда, зонт, ветер
	Как доехать?	транспорт, такси, каршэринг
пользователь в пятницу или субботу вечером собирается куда-то поехать	Куда поехать?	ресторан, кафе, бар, кальянная
	Что надеть?	погода, одежда, зонт, ветер.
	Как доехать?	транспорт, такси, каршэринг





**Сервисы
умного города**



Связывание



**Запросы
пользователей**



Варианты реализации

1. Системы поддержки принятия решений
2. Вопросно-ответные системы
3. Виртуальный ассистент
4. Сервис рекомендаций

Вопросно-ответные системы

(Question answering systems, QAS)

Примеры QAS

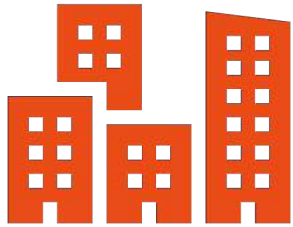
-QALD 1	-QALD 3	Intui3	-QALD 6	gAnswer	Start
FREYA	squal2sparql	ISOFT	CANaLI	TeBaQA	TBSL
poweraqua	CASIA	RO_FII	PersionQA	QASystem	LODQA
SWIP	Scalewelis	-QALD 5	UTQA	-Miscellaneous	AskNow
-QALD 2	RTV	APEQ	KGWAnswer	NLSearch	Platypus
Alexandria	Intui2	QAnswer	NbFramework	OKBQA	Quepy
SemSeK	-QALD 4	SemGraphQA	UIQA	DEANNA	SorokinQA
MHE	Xser	YodaQA	-QALD 9	FRANKENSTEIN	questIO
QAKIS	gAnswer	HAWK	WDAqua-core1	SINA	QUINT

Типы QAS

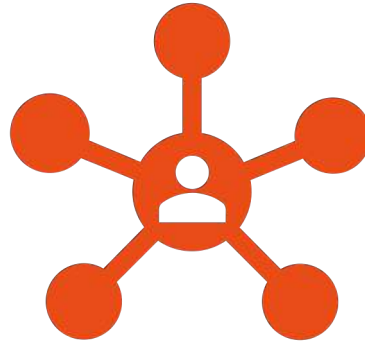
1. С использованием машинного обучения
2. С использованием семантического веба

Типы QAS

1. С использованием машинного обучения
2. С использованием семантического веба



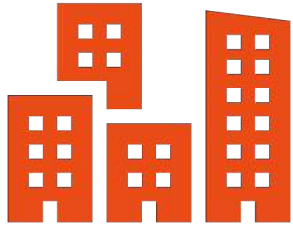
**Сервисы
умного города**



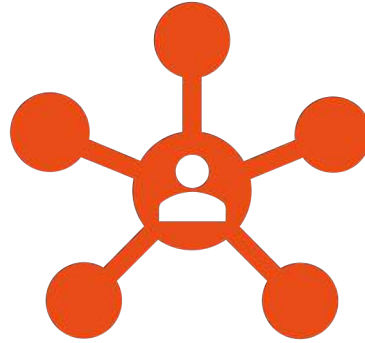
Связывание



**Запросы
пользователей**



**Сервисы
умного города**



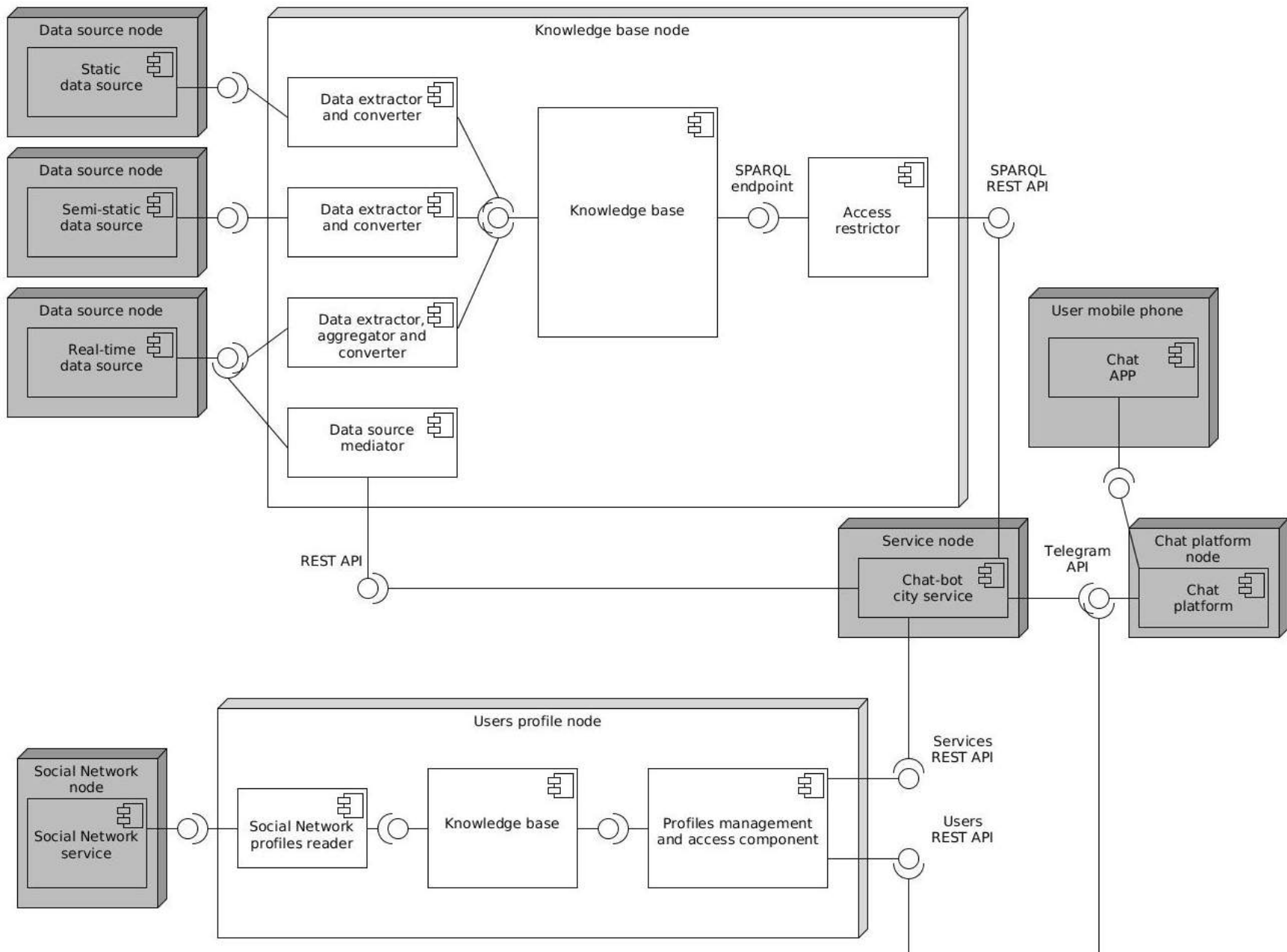
Связывание

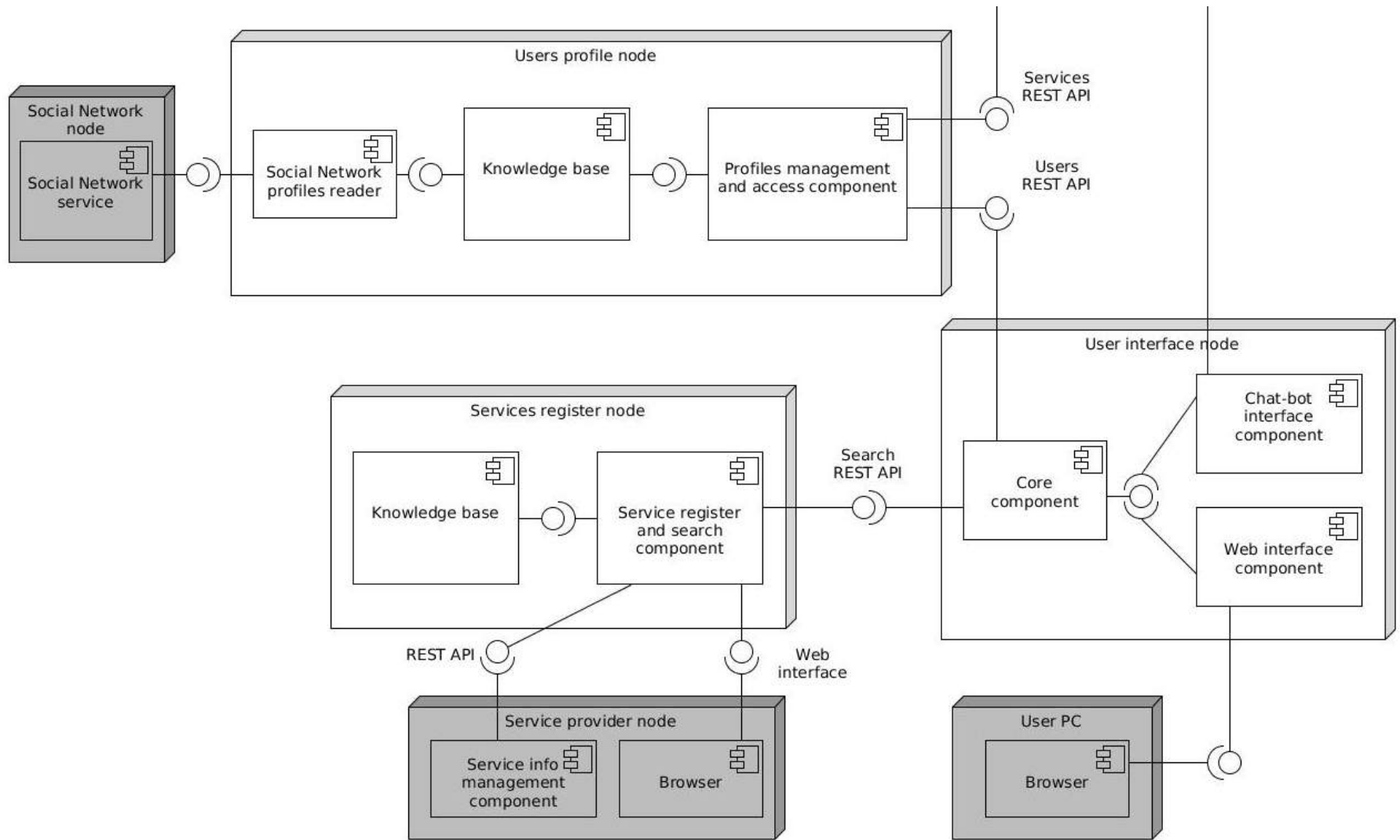


**Запросы
пользователей**

Как реализовать?

Архитектура





Алгоритм выполнения запросов на естественном языке

Описание алгоритма

1. Берем базу запросов из **facebook** с сущностями с **freebase**
2. Делаем маппинг **freebase** сущностей в **wikidata**
3. Подготавливаем данные (чистим, переводим, лемматизируем, убираем стопслова)
4. Обрабатываем данные (генерируем запросы SPARQL в wikidata)
5. Сохраняем полученные результаты для выявления оптимальной стратегии поиска связанных сущностей
6. Вычисляем точность алгоритмов для каждого варианта представления строк и для разного числа результатов

Примеры работы алгоритма выполнения запросов на естественном языке

1. Ввод запроса: **Как добраться до аэропорта**

1. Ввод запроса: **Как добраться до аэропорта**
2. Определение сущностей:
 1. Добраться → сущность **Маршрут** (id **7732**)
 2. Аэропорт → сущность **Аэропорт** (id **4380**)

1. Ввод запроса: **Как добраться до аэропорта**
2. Определение сущностей:
 1. Добраться → сущность **Маршрут** (id **7732**)
 2. Аэропорт → сущность **Аэропорт** (id **4380**)
3. Определение сервиса умного города для сущности **Маршрут**:
 1. **Сервис построения маршрутов**
 2. **Сервис такси**

1. Ввод запроса: **Как добраться до аэропорта**
2. Определение сущностей:
 1. Добраться → сущность **Маршрут** (id **7732**)
 2. Аэропорт → сущность **Аэропорт** (id **4380**)
3. Определение сервиса умного города для сущности **Маршрут**:
 1. **Сервис построения маршрутов**
 2. **Сервис такси**
4. Определение параметров сервиса

1. Ввод запроса: **Как добраться до аэропорта**
2. Определение сущностей:
 1. Добраться → сущность **Маршрут** (id **7732**)
 2. Аэропорт → сущность **Аэропорт** (id **4380**)
3. Определение сервиса умного города для сущности **Маршрут**:
 1. **Сервис построения маршрутов**
 2. **Сервис такси**
4. Определение параметров сервиса
5. Выполнение запроса

Результаты работы алгоритмы

	top1	top3	top5	top10	top30	top50	top100
nls	20%	25%	27%	32%	36%	39%	45%
nls lemmas	29%	33%	36%	40%	43%	47%	51%
nls cutted begin	20%	23%	26%	31%	37%	40%	44%
nls cutted all	20%	21%	24%	26%	35%	37%	42%
nls cutted begin by lemmas	23%	25%	27%	34%	40%	42%	46%
nls cutted all by lemmas	17%	20%	22%	24%	29%	31%	35%
nls cutted lemmas, processed	14%	17%	19%	23%	27%	29%	33%
nls cutted start, processed	15%	20%	22%	26%	29%	32%	38%
nls cutted all, processed	12%	14%	16%	19%	22%	24%	28%
nls cutted start by ru lemmas, processed	10%	13%	15%	19%	22%	23%	26%
nls cutted all by ru lemmas, processed	9%	11%	13%	16%	19%	20%	24%

ИТОГИ

Платформа

- Сделали прототип системы
- Архитектура ОК
- Очень **гибко** (важно для научных решений)

Алгоритм выполнения запросов на естественном языке

- **Низкая точность**
- Машинное обучение (например, деревья решений) + описанные грамматики для сервисов лучше подходят для бизнеса (Сири, Алиса, ...)



Евгений Липкин

Университет ИТМО

E-mail: elipkin@itmo.ru