

MDM, который ничего не хранит

Как сопоставлять данные, не забирая их к себе

Юрий Горынцев
Директор продукта



Коротко про меня

- Я разработчик
- Первые деньги за свой код получил в 7 классе
- Работал над приложениями для X5, МВидео, Мега
- Бывал в Яндексе
- С головой ушел в качество данных
- Теперь Product Owner и лид DQ продукта

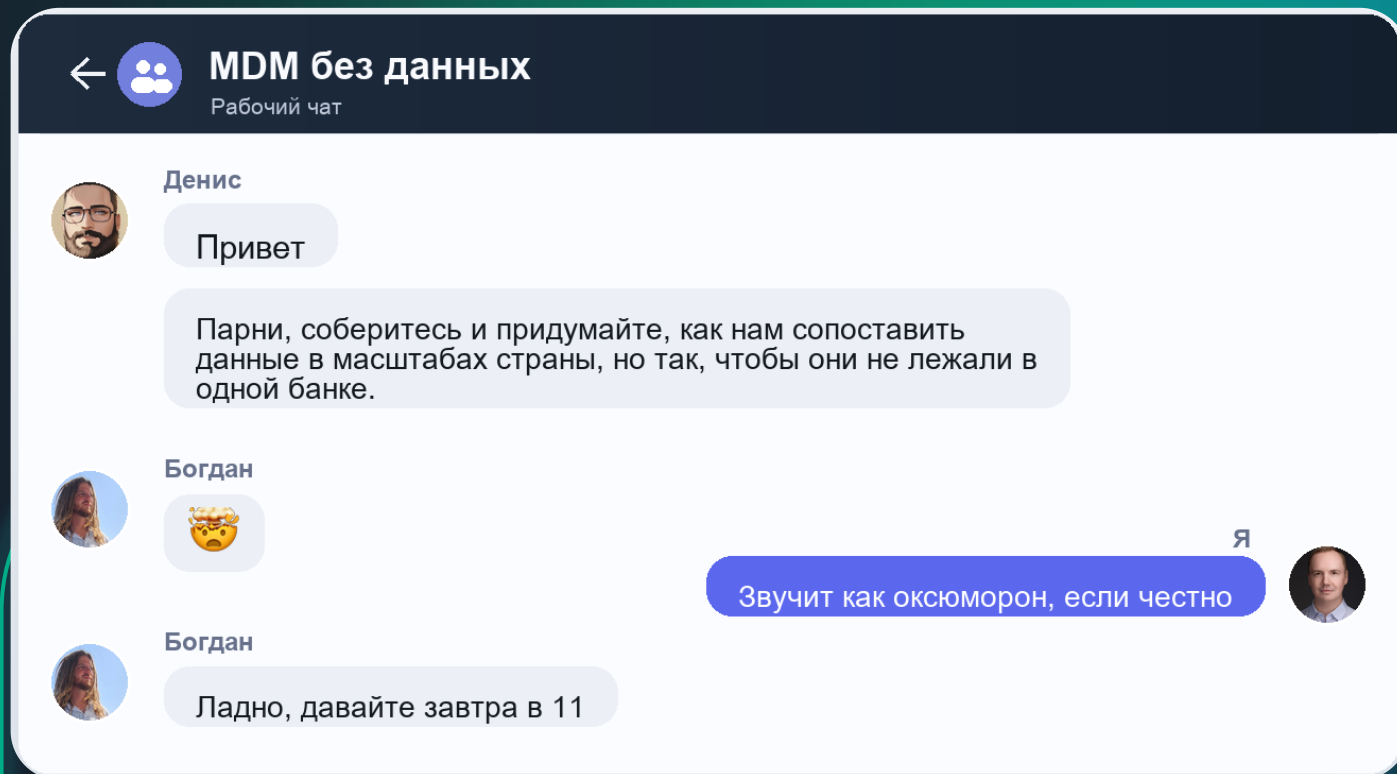
О чем сегодня поговорим

- Про нашу задачу
- Как работает «классический» MDM и почему он нам не подходит
- Что такое персональные данные
- Первая версия алгоритма
- Подробно разберем итоговую версию

Постановка задачи

Нужно сопоставить данные о людях из «витрин» для возможности выполнения Ad-hoc аналитических запросов

Нельзя допустить концентрации ПДн в одной базе



Что у нас есть



СМЭВ предоставляет

- Защищенный транспорт
- Кросс-витринные запросы с возможностью регулирования доступов

Вопросы к постановке

- Откуда родилось требование про «одну банку»?
- Почему у меня такая реакция на это требование?
- Если есть кросс-витринные запросы, то может и не нужно собирать?

Классический MDM

Что под капотом

1. Загрузка

1. пакетно
- потоково
- через API

2. Стандартизация

2. стандартизация адресов;
- унификация телефонов;
- приведение регистра;

3. DQ

3. обязательные поля;
- контрольные суммы;
- дата допустима;

4. Поиск и сопоставление

4. Поговорим об этом на следующих слайдах

5. Построение Golden Record

5. самый свежий;
- самый доверенный;
- самый полный;
- ручное подтверждение

6. Публикация

6. 1. изменился адрес;
2. послали обновление;
3. системы обновили;

Сопоставление

Прямой перебор – очень дорого: $N * (N-1) / 2$

Blocking

- Одинаковый ИНН
- ДР + первая буква фамилии
- Одинаковый телефон
- Одинаковый фонетический ключ
- ...

Все зависит от правил

Scoring

Атрибут	Метод	Вес
Фамилия	Jaro-Winkler	0.25
Имя	Jaro-Winkler/инициал	0.15
Отчество	Jaro-Winkler/инициал	0.10
ДР	exact	0.25
Телефон	Normalized exact	0.25

≥ 0.95 – автослияние

$0.80 - 0.95$ – на проверку оператору

< 0.80 – не сливаем

Нечеткий поиск

- Damerau-Levenshtein distance
- Jaro-Wikler
- N-gram similarity
- Token-based
 - Token sort ratio
 - Token set ratio
 - Jaccard Similarity (пересечение/объединение)
 - Cosine similarity с TF-IDF

Фонетические алгоритмы

- Double Metaphone
- Daitch-Mokotoff Soundex
- Beider-Morse

И вот мы получили ее – эталонную базу обо всем



Определяемся с терминами и ландшафтом

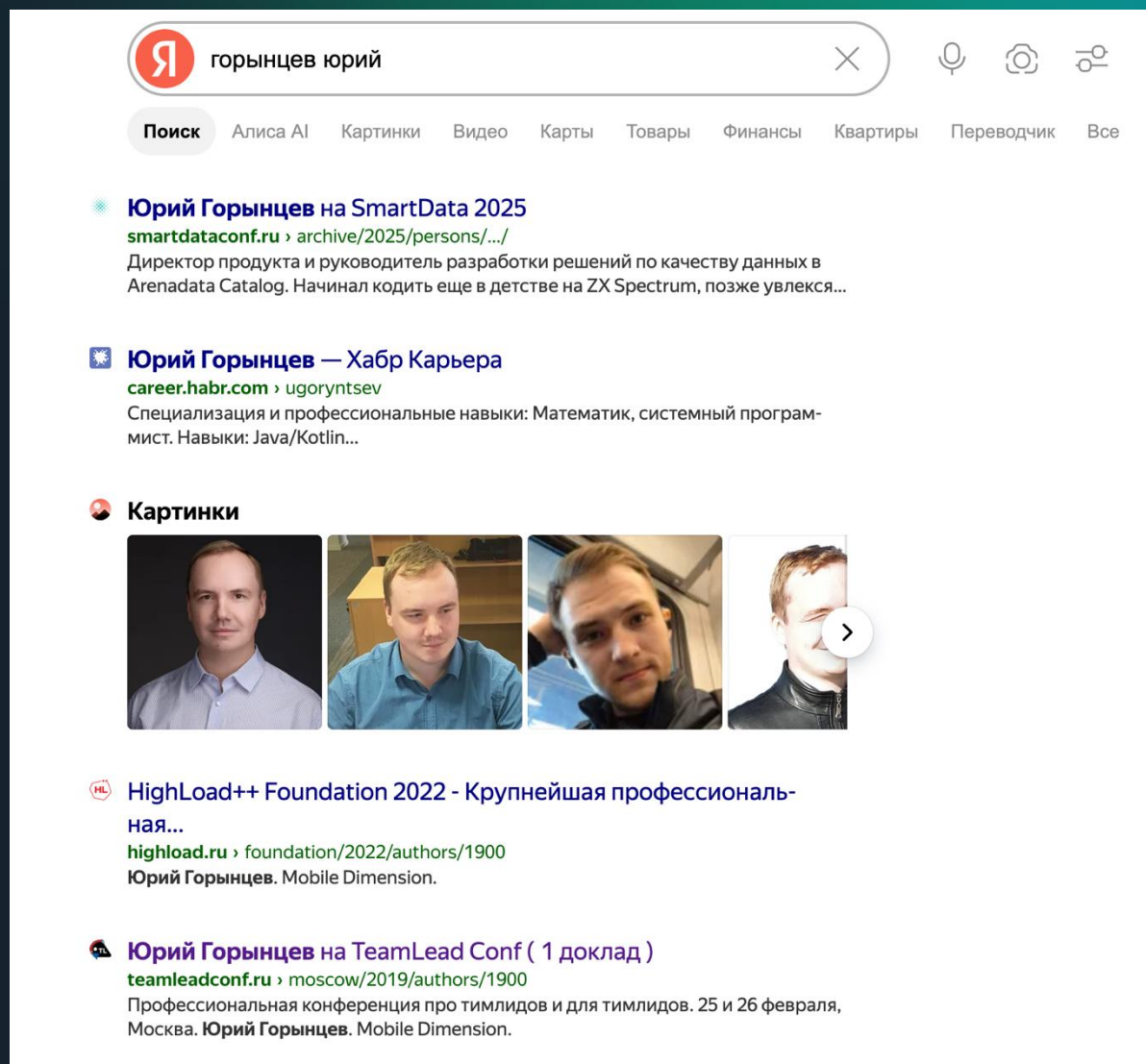
Что такое персональные данные

Официальное определение:

Персональные данные - любая информация, относящаяся к прямо или косвенно определенному или определяемому физическому лицу (субъекту персональных данных); ⁽¹⁾

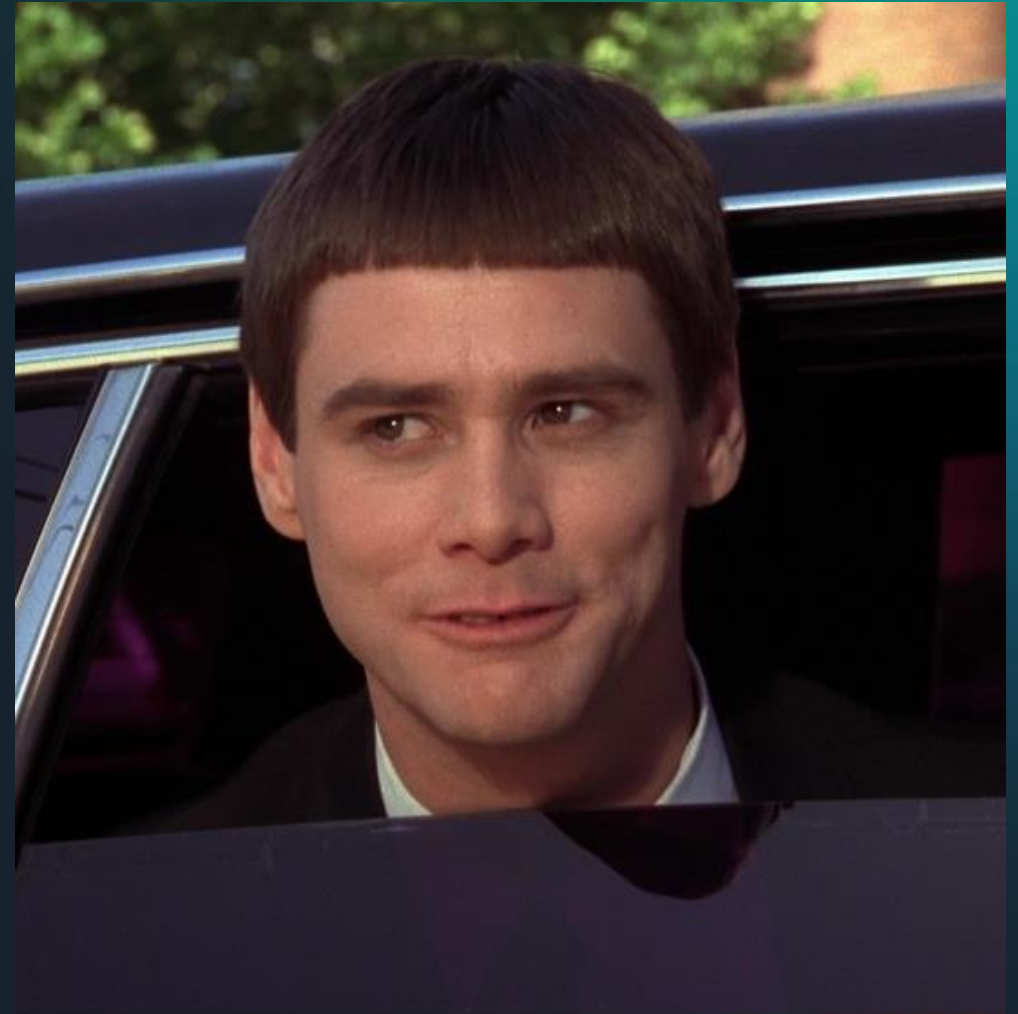
Общепринятое понимание:

Персональные данные - это, например, **ФИО + номер телефона**



Первые мысли

- А давайте все зашифруем, а ключ выкинем?
- Тогда сравнивать не получится!
- Тогда захешируем?!
- Хэш от персональных данных – это персональные данные! (2)



И действительно...

Перечень методов обезличивания закрытый: ⁽³⁾

- Введение идентификаторов
- ~~Изменение состава или семантики~~
- Декомпозиция
- Перемешивание



Разделяй и властвуй

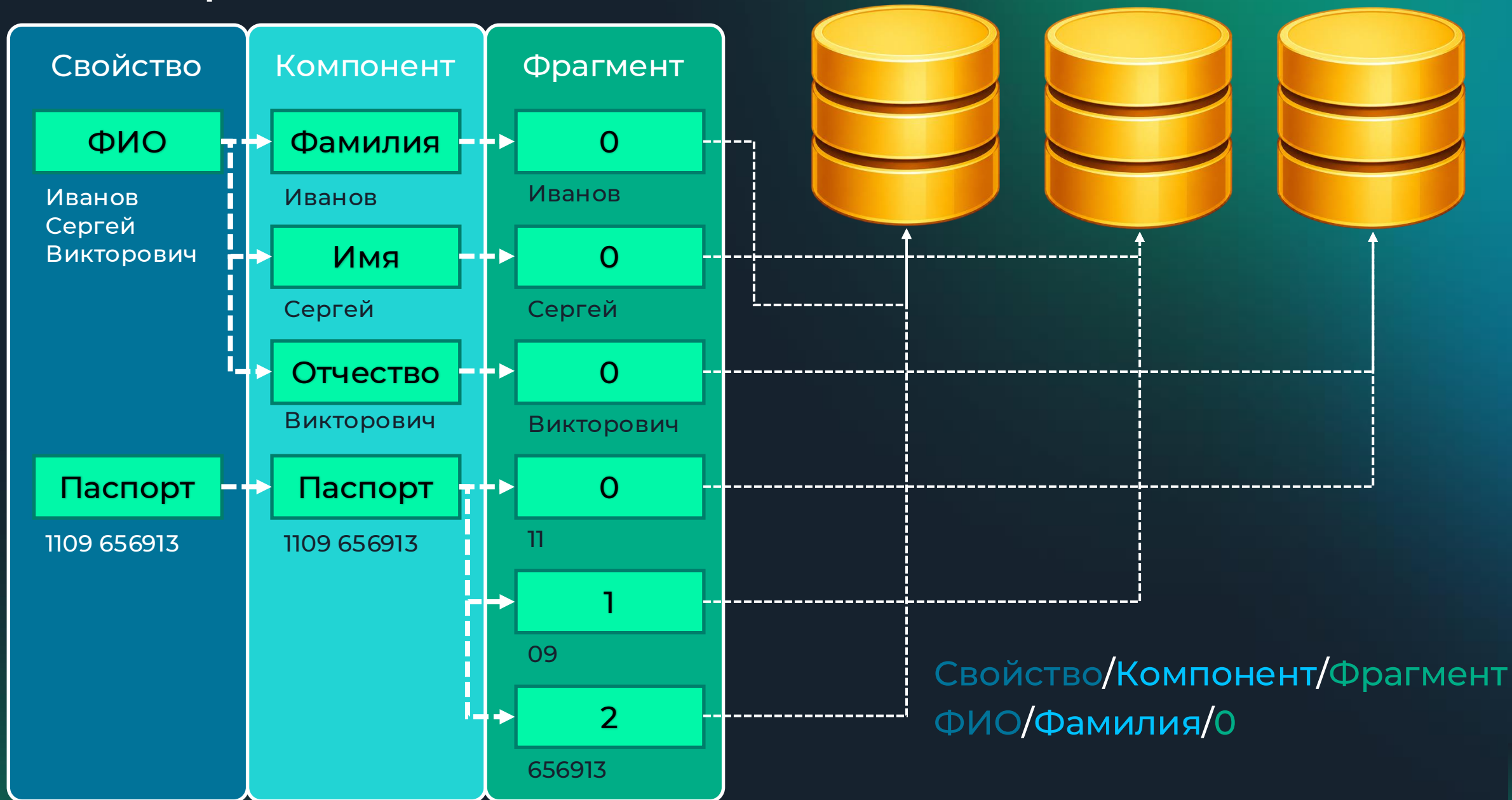


Горынцев Юрий Сергеевич – десятки, но:
Горынцев – тысячи
Юрий – сотни тысяч
Сергеевич – миллионы

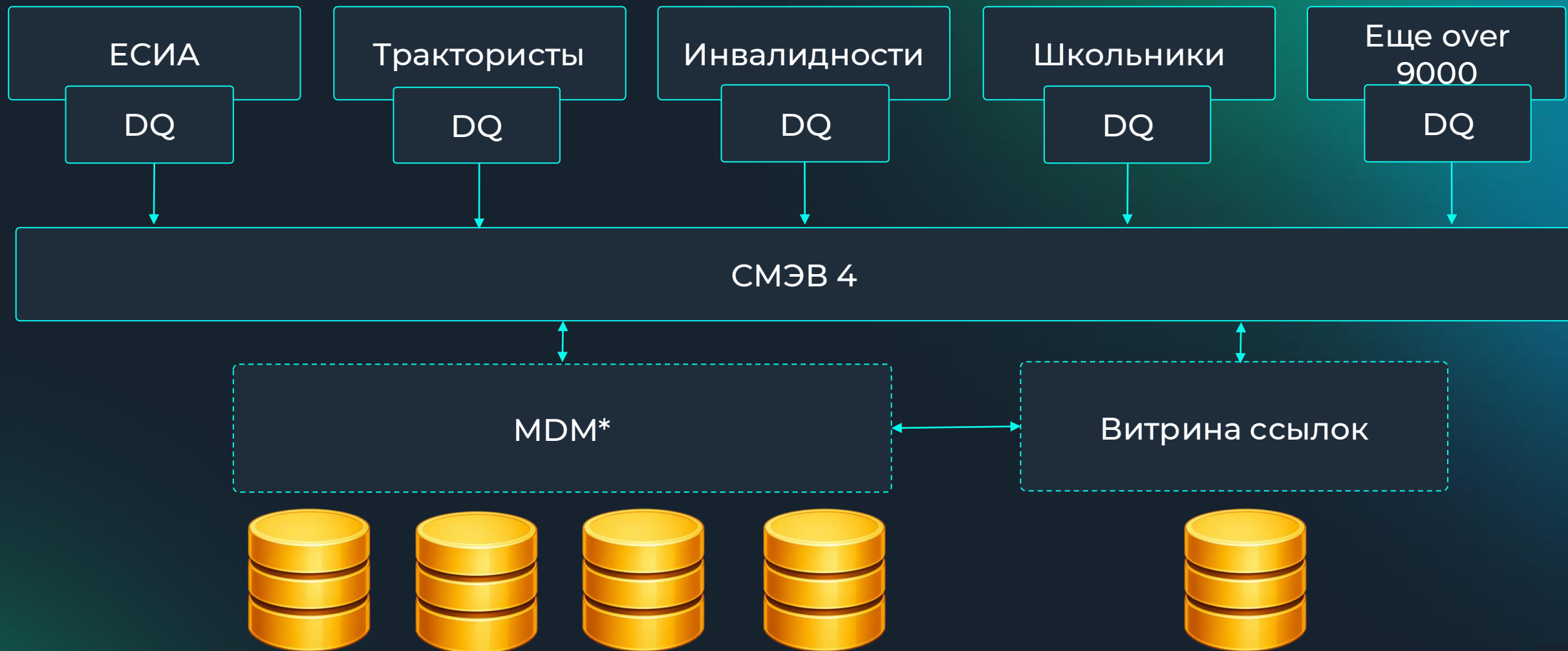
Структура паспорта: **RRYY NNNNNN**
RR – код региона ОКАТО
YY – год выдачи +/- 1 год
NNNNNN – шестизначный номер

Горынцев, с номером ОКАТО 11 – сотни
Горынцев, с годом выдачи 09 – десятки

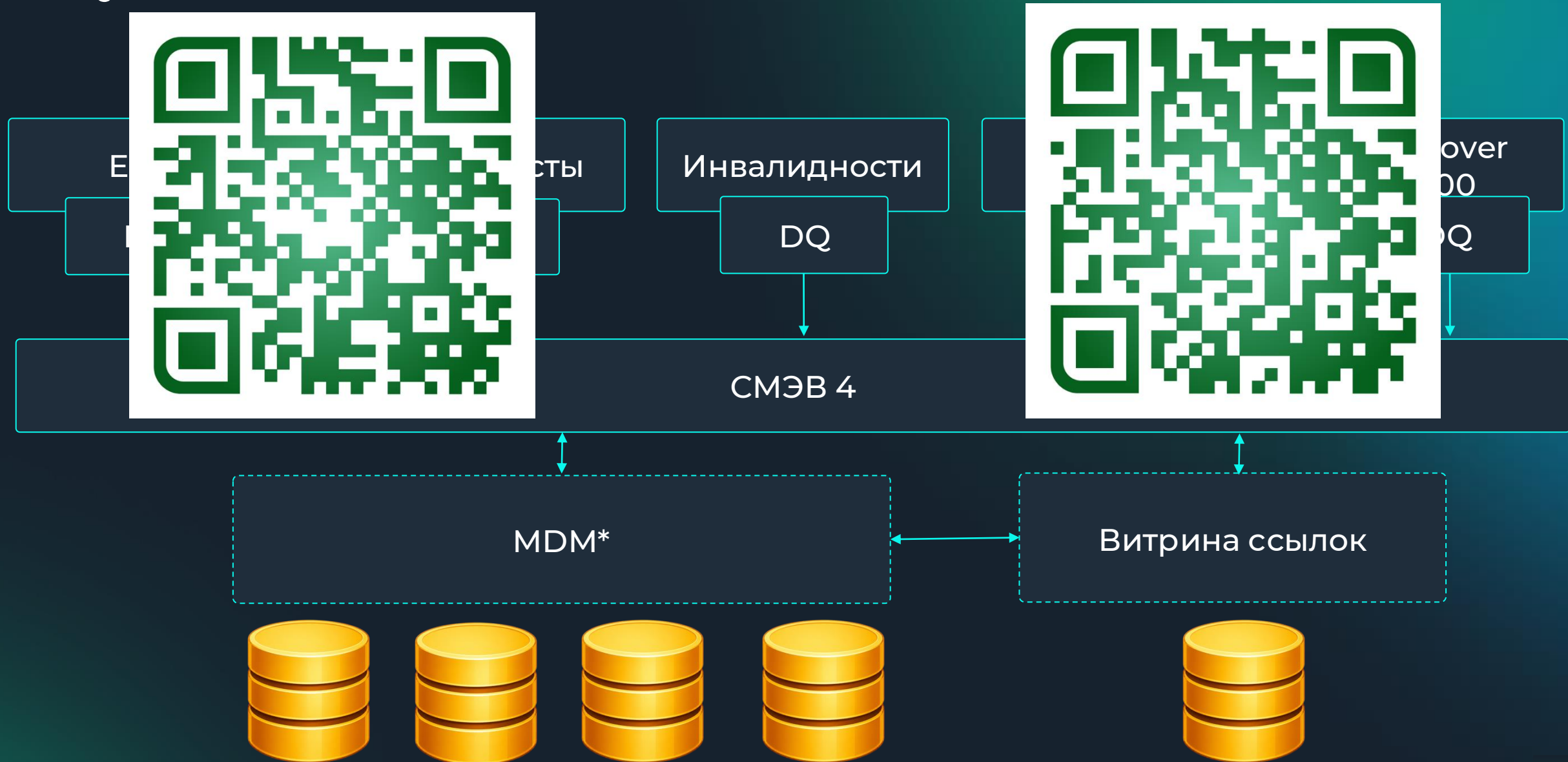
Как конкретно делим?



Что у нас есть

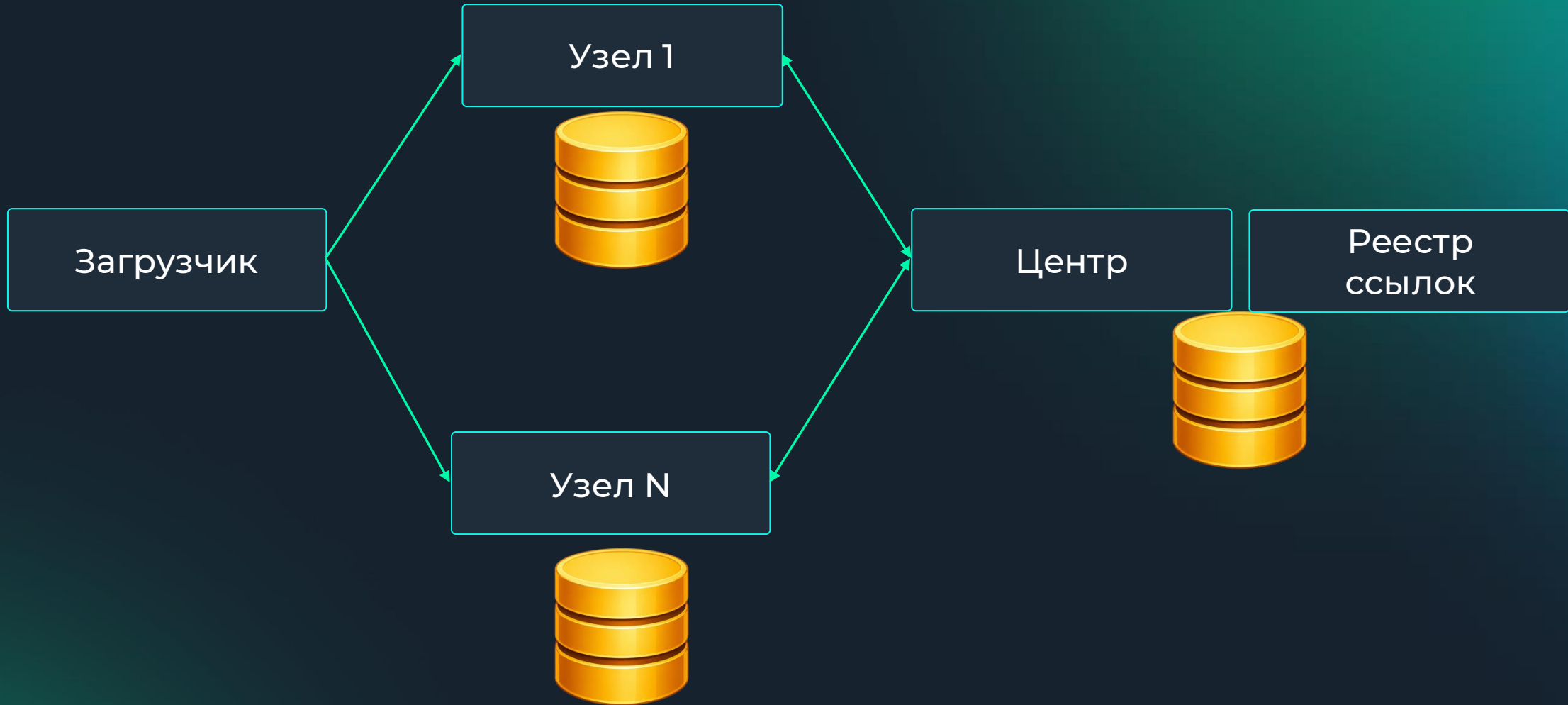


Что у нас есть

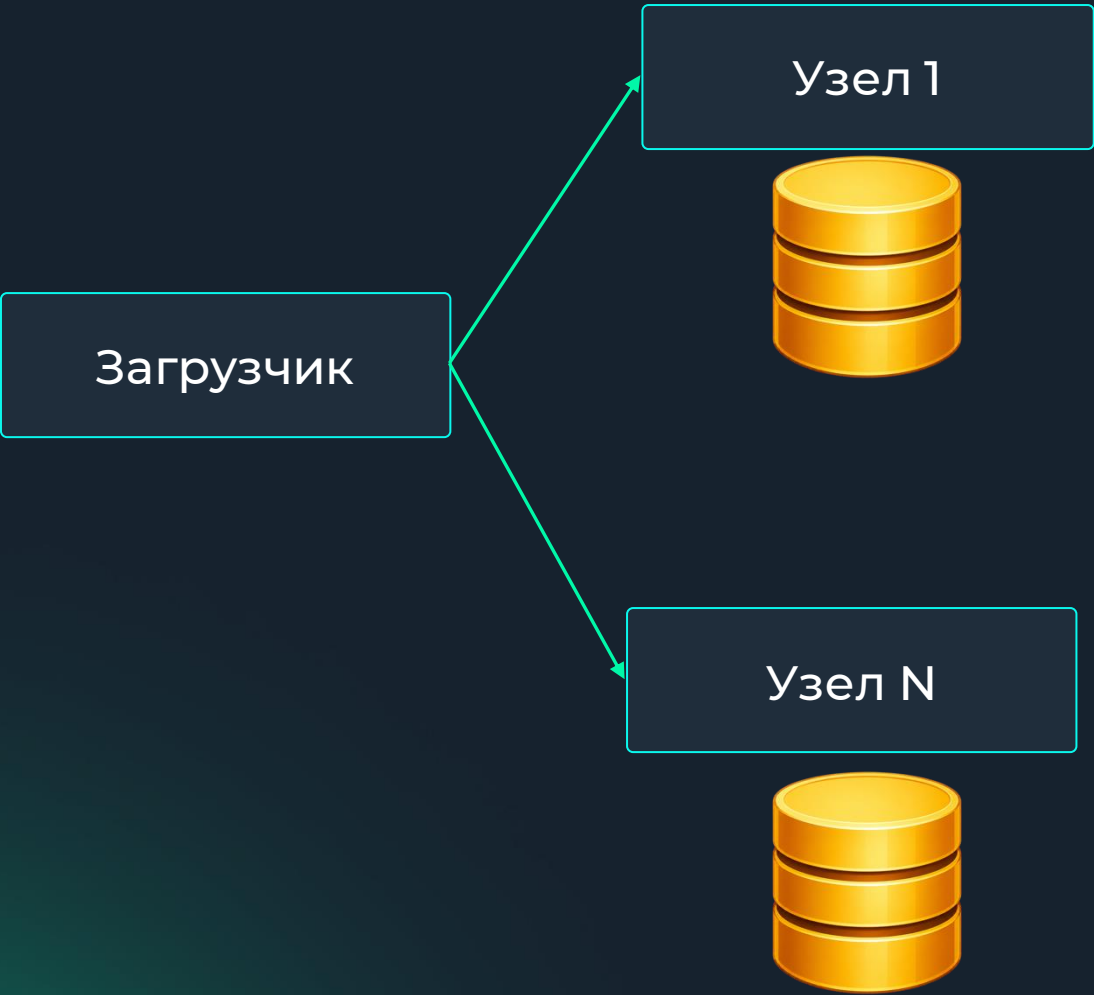


Первая версия сопоставления

Архитектура



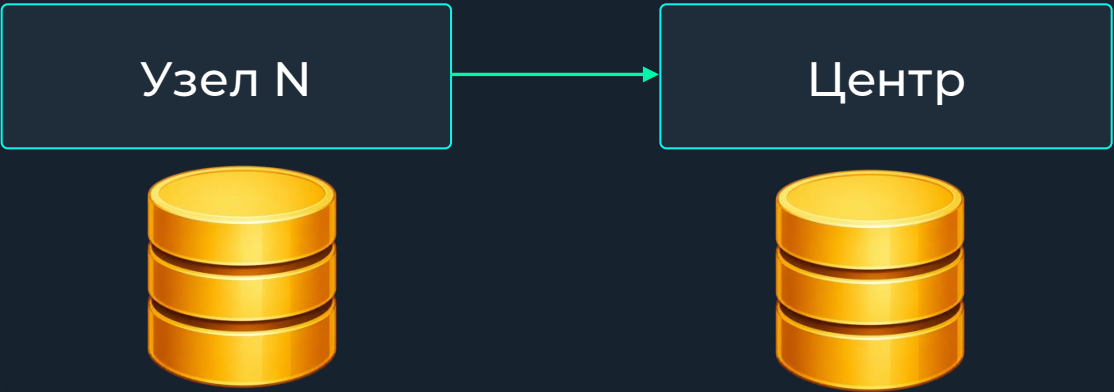
1. Узловое обновление



Имя		Тип	Описание
sourceId		String	Идентификатор источника Не может содержать пустое значение или значение "-"
sourceCardId		String	Идентификатор карточки в источнике
fragments		Array • Object	Массив фрагментов, относящийся к данной карточке
	name	String	Имя фрагмента, например "ФИО/ФАМИЛИЯ/0"
	val	String	Значение фрагмента. Отсутствующее значение может быть представлено одним из следующих вариантов: • JSON-свойство val отсутствует • JSON-свойство val имеет значение null • JSON-свойство val равно пустой строке.

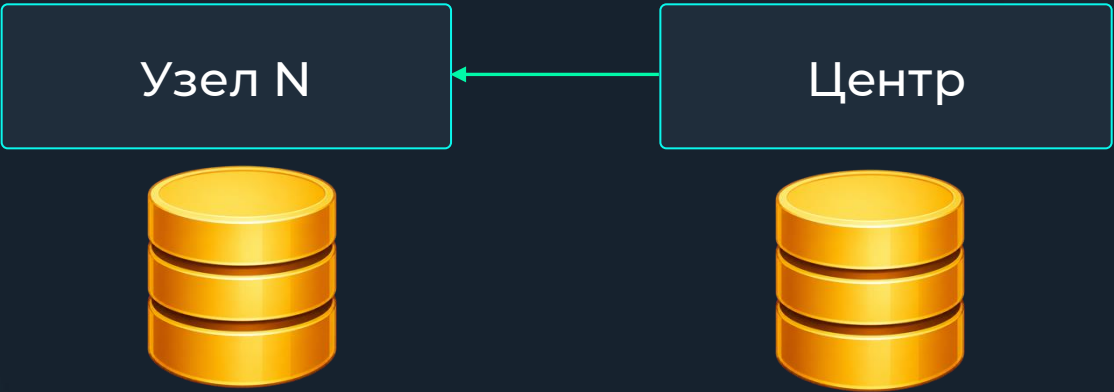
2 Центральное обновление

Имя свойства	Тип JSON	Описание
nodeId	String	Идентификатор Узла
sourceId	String	Идентификатор источника
cardId	String	Идентификатор исходной карточки в источнике



3. Поисковый запрос

Имя свойства	Тип JSON	Описание
ruleId	String	Идентификатор поискового правила
sourceId	String	Идентификатор источника
sourceCardId	String	Идентификатор исходной карточки в источнике



Правила поиска

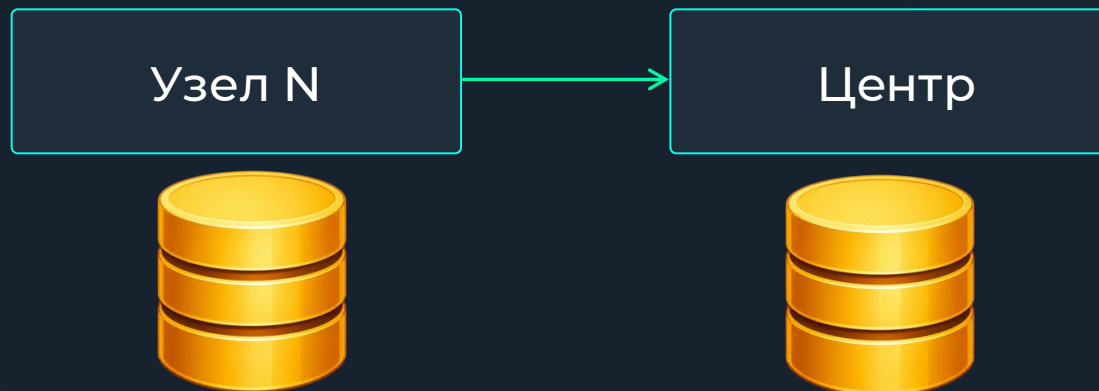
Приоритет	Описание	ФИО	ДР	ПАСПОРТ	СНИЛС	ИНН
1	ФИО, даты рождения, паспорта РФ (серия, номер), СНИЛС и ИНН совпадают					
2	ФИО, даты рождения, паспорта РФ (серия, номер), СНИЛС совпадают, ИНН не противоречит					
3	ФИО, даты рождения, паспорта РФ (серия, номер), ИНН совпадают, СНИЛС не противоречит					
4	ФИО, даты рождения, СНИЛС и ИНН совпадают, паспорта РФ (серия, номер) не противоречит					
5	ФИО, даты рождения и паспорта РФ (серия, номер) совпадают, СНИЛС и ИНН не противоречит					
6	ФИО, даты рождения и СНИЛС совпадают, паспорта РФ (серия, номер) и ИНН не противоречит					
7	ФИО, даты рождения и ИНН совпадают, паспорта РФ (серия, номер) и СНИЛС не противоречит					

Легенда

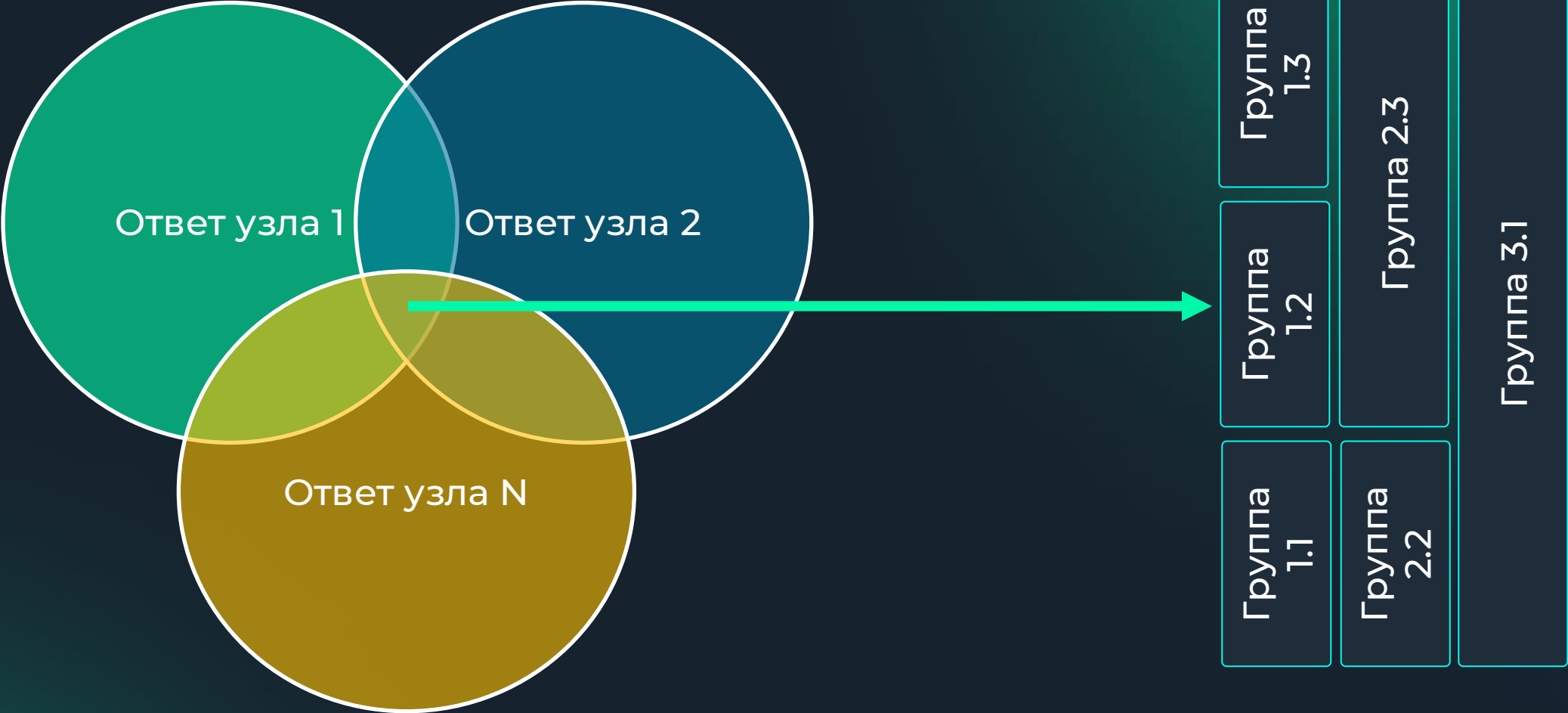
	совпадает
	не противоречит

4. Ответ поиска

Имя свойства		Тип JSON	Описание
ruleId		String	Идентификатор поискового правила
cardIds		Array • Object	Массив карточек, найденных по правилу
	sourceId	String	Идентификатор источника
	sourceCardId	String	Идентификатор исходной карточки в источнике



5. Результат



Минутка рефлексии

Плюсы и минусы

- + Прозрачность хранения сопоставлений
- + Одна карточка сопоставлений может содержать в себе несколько карточек
- - Неразрешимое противоречие относительно группировки

Открытые вопросы

- Как обрабатывать изменения/удаления?
- Что делать с транзитивным сопоставлением?
- Как разрешать конфликты?
- Как работать с множественными атрибутами?

Нефункциональные результаты

- 30-40 сопоставлений/сек
- А нужно 7 000

Эволюция алгоритма

Расширяем методы сопоставления

Метод сопоставления	Значение компонента в карточке №1	x ¹	x ²	∅ ³	x ⁴
	Значение компонента в карточке №2	y	∅	∅	x
REQ_EQUAL	Равенство обязательного компонента	0	0	0	1
OPT_EQUAL	Равенство опционального компонента	0	0	1	1
OPT_CONSISTENT	Непротиворечие опционального компонента	0	1	1	1

Компонент	Метод сопоставления
ФИО/Фамилия	REQ_EQUAL
ФИО/Имя	REQ_EQUAL
ФИО/Отчество	OPT_EQUAL
ДР/ДР	REQ_EQUAL
СНИЛС/СНИЛС	OPT_CONSISTENT

Карточка	Фамилия	Имя	Отчество	ДР	СНИЛС
A	Иванов	Александр	Борисович	15.03.1989	112-233-445 95
	Иванов	Александр			
B	Иванов	Александр		15.03.1989	
C	Иванов	Александр		15.03.1989	112-288-445 39
	Иванов	Александр	Борисович		
D	Иванов	Александр	Борисович	15.03.1989	112-288-445 39
E	Иванов	Александр			112-288-445 39
	Иванов	Александр	Борисович		

Вводим поисковые теги

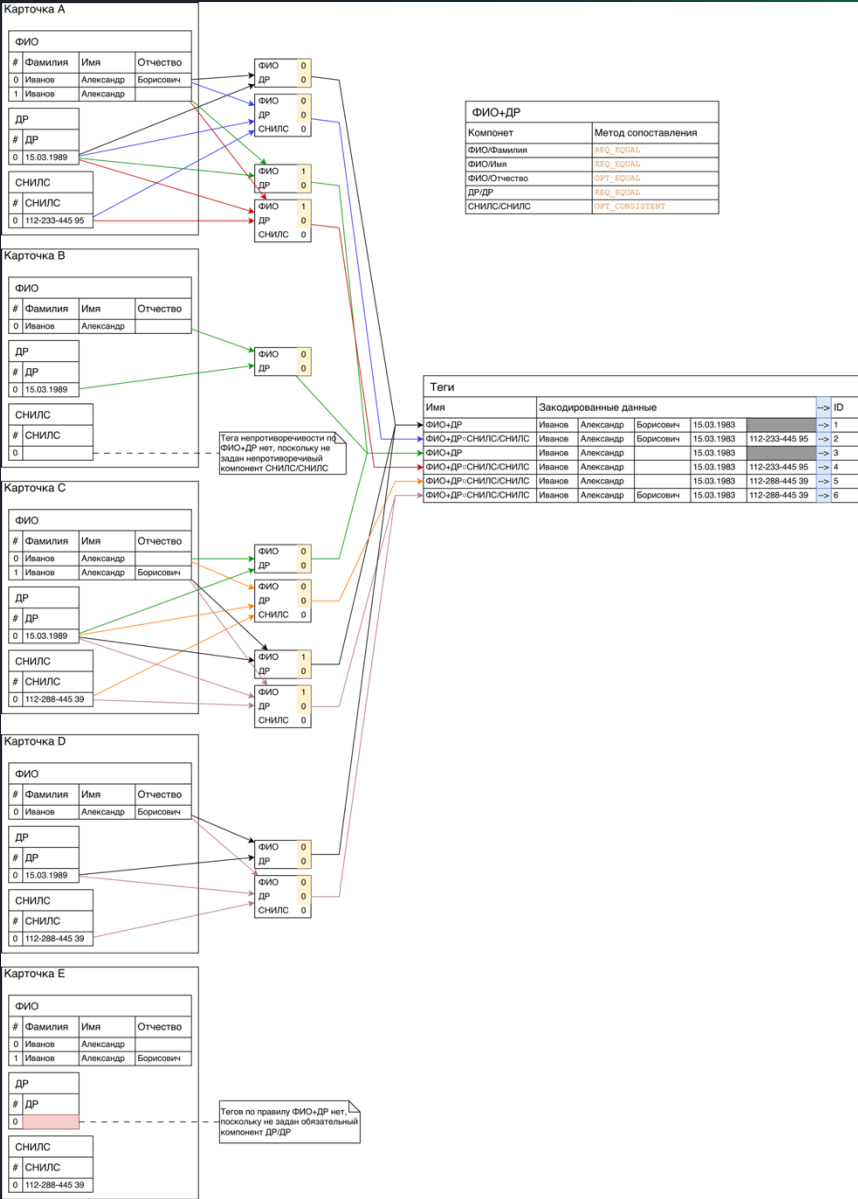
Основная идея – преобразовать ключевые характеристики документов в **перечислимые** типы

У нас документы – это карточки, а характеристики – набор компонентов

Теги 2 видов:

- **Каноничные** – наборы компонентов, которые должны совпадать (**REQ_EQUAL** и **OPT_EQUAL**)
- **Непротиворечивости** – объединение значений **Каноничного** и еще одного значения, которое не противоречит (**OPT_CONSISTENT**)

Вводим поисковые теги. Пример



Карточка А

ФИО

#	Фамилия	Имя	Отчество
0	Иванов	Александр	Борисович
1	Иванов	Александр	

ДР

#	ДР
0	15.03.1989

СНИЛС

#	СНИЛС
0	112-233-445 95

ФИО 0

ДР 0

ФИО 0

ДР 0

СНИЛС 0

ФИО 1

ДР 0

ФИО 1

ДР 0

СНИЛС 0

ФИО+ДР	
Компонет	Метод сопоставления
ФИО/Фамилия	REQ_EQUAL
ФИО/Имя	REQ_EQUAL
ФИО/Отчество	OPT_EQUAL
ДР/ДР	REQ_EQUAL
СНИЛС/СНИЛС	OPT_CONSISTENT

Карточка В

ФИО

#	Фамилия	Имя	Отчество
0	Иванов	Александр	

ДР

#	ДР
0	15.03.1989

СНИЛС

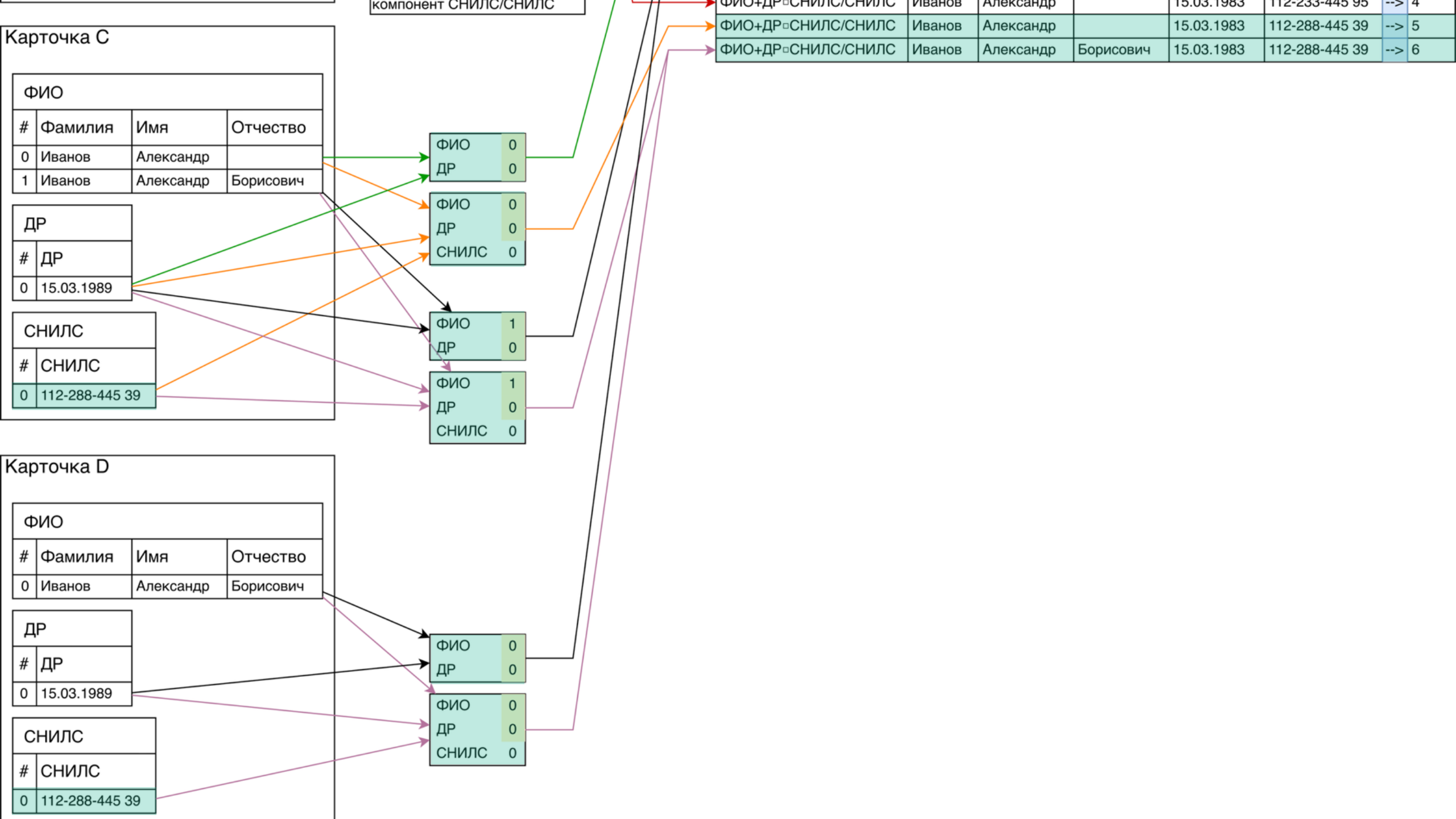
#	СНИЛС
0	

ФИО 0

ДР 0

Тега непротиворечивости по ФИО+ДР нет, поскольку не задан непротиворечивый компонент СНИЛС/СНИЛС

Теги							
Имя		Закодированные данные					
						-->	ID
ФИО+ДР		Иванов	Александр	Борисович	15.03.1983		--> 1
ФИО+ДР□СНИЛС/СНИЛС		Иванов	Александр	Борисович	15.03.1983	112-233-445 95	--> 2
ФИО+ДР		Иванов	Александр		15.03.1983		--> 3
ФИО+ДР□СНИЛС/СНИЛС		Иванов	Александр		15.03.1983	112-233-445 95	--> 4
ФИО+ДР□СНИЛС/СНИЛС		Иванов	Александр		15.03.1983	112-288-445 39	--> 5



Карточка D

ФИО

#	Фамилия	Имя	Отчество
0	Иванов	Александр	Борисович

ДР

#	ДР
0	15.03.1989

СНИЛС

#	СНИЛС
0	112-288-445 39

СНИЛС 0

ФИО 0

ДР 0

ФИО 0

ДР 0

СНИЛС 0

Карточка E

ФИО

#	Фамилия	Имя	Отчество
0	Иванов	Александр	
1	Иванов	Александр	Борисович

ДР

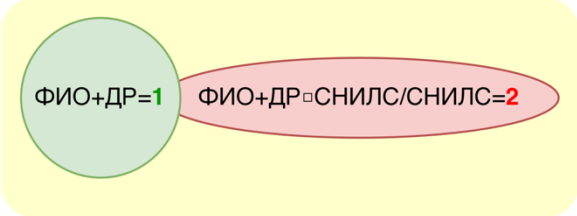
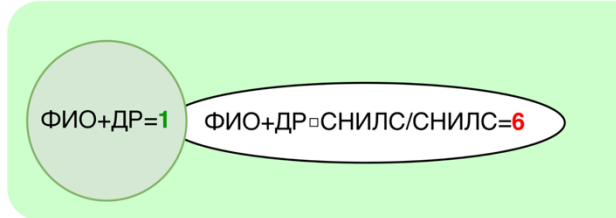
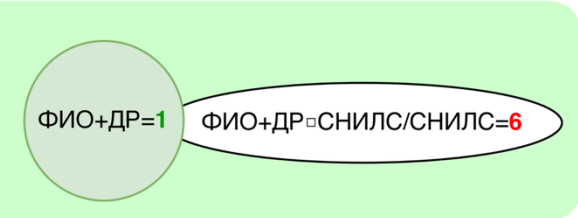
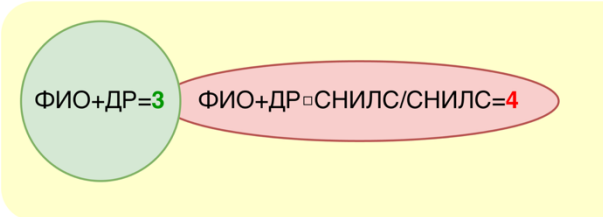
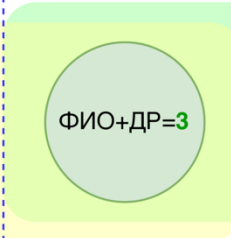
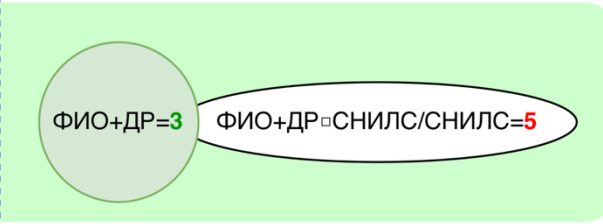
#	ДР
0	

СНИЛС

#	СНИЛС
0	112-288-445 39

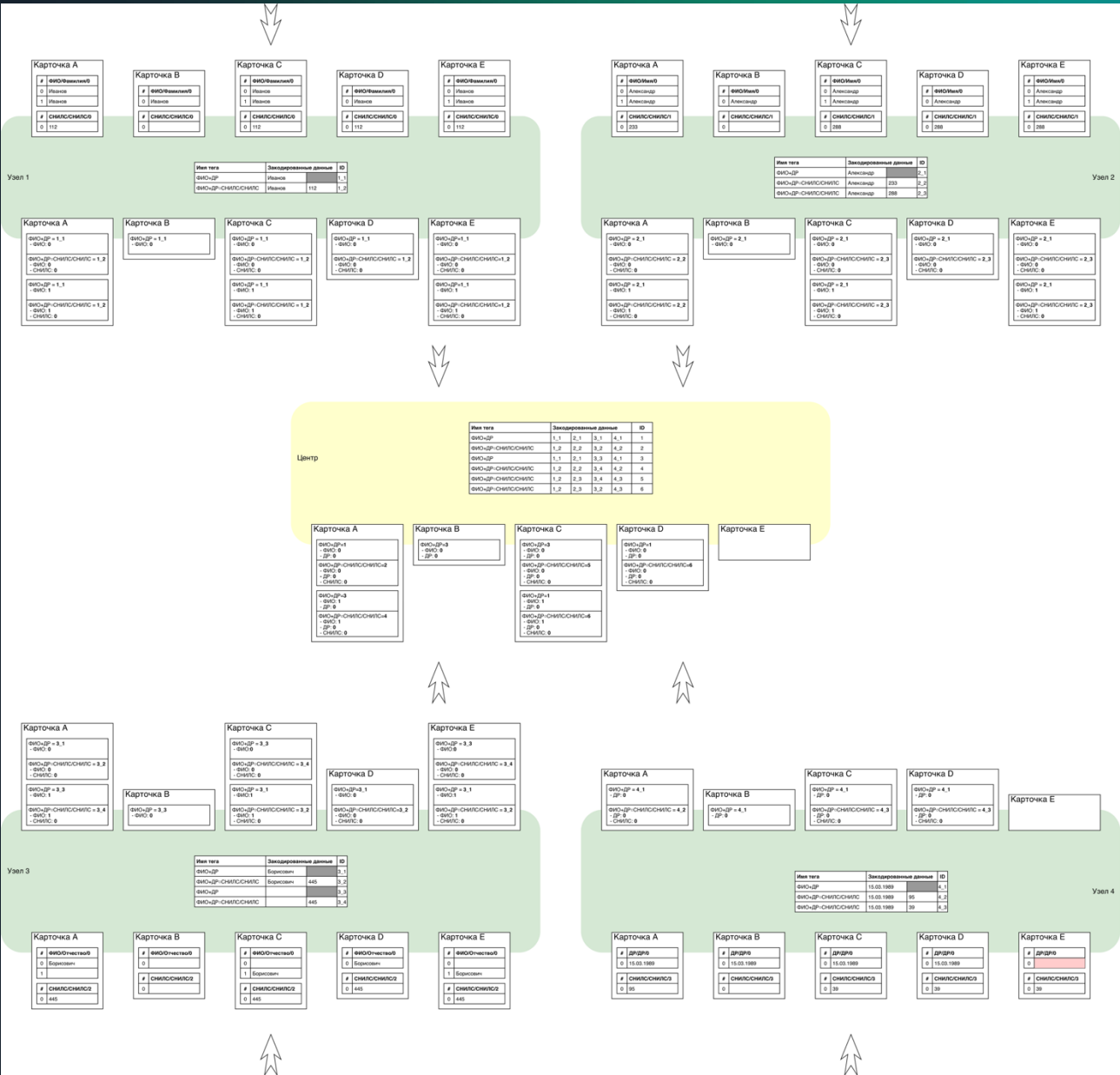
Тегов по правилу ФИО+ДР нет,
поскольку не задан обязательный
компонент ДР/ДР

Сопоставление в терминах тегов

Карточка А	Карточка В	Карточка С	Карточка D	Карточка Е
				
				

Распределенные поисковые теги

Фрагмент	Узел	Пример
ФИО/Фамилия/0	Узел 1	Иванов
ФИО/Имя/0	Узел 2	Александр
ФИО/Отчество/0	Узел 3	Борисович
ДР/ДР/0	Узел 4	15.03.1989
СНИЛС/СНИЛС/0	Узел 1	112-288-445 39
СНИЛС/СНИЛС/1	Узел 2	112-288-445 39
СНИЛС/СНИЛС/2	Узел 3	112-288-445 39
СНИЛС/СНИЛС/3	Узел 4	112-288-445 39



Карточка А

#	ФИО/Фамилия/0
0	Иванов
1	Иванов

#	СНИЛС/СНИЛС/0
0	112

Карточка В

#	ФИО/Фамилия/0
0	Иванов

#	СНИЛС/СНИЛС/0
0	

Карточка С

#	ФИО/Фамилия/0
0	Иванов
1	Иванов

#	СНИЛС/СНИЛС/0
0	112

Карточка D

#	ФИО/Фамилия/0
0	Иванов

#	СНИЛС/СНИЛС/0
0	112

Карточка Е

#	ФИО/Фамилия/0
0	Иванов
1	Иванов

#	СНИЛС/СНИЛС/0
0	112

Узел 1

Имя тега	Закодированные данные		ID
ФИО+ДР	Иванов		1_1
ФИО+ДР+СНИЛС/СНИЛС	Иванов	112	1_2

Карточка А

ФИО+ДР = 1_1

- ФИО: 0

ФИО+ДР+СНИЛС/СНИЛС = 1_2

- ФИО: 0

- СНИЛС: 0

ФИО+ДР = 1_1

- ФИО: 1

ФИО+ДР+СНИЛС/СНИЛС = 1_2

- ФИО: 1

- СНИЛС: 0

Карточка В

ФИО+ДР = 1_1

- ФИО: 0

Карточка С

ФИО+ДР = 1_1

- ФИО: 0

ФИО+ДР+СНИЛС/СНИЛС = 1_2

- ФИО: 0

- СНИЛС: 0

ФИО+ДР = 1_1

- ФИО: 1

ФИО+ДР+СНИЛС/СНИЛС = 1_2

- ФИО: 1

- СНИЛС: 0

Карточка D

ФИО+ДР = 1_1

- ФИО: 0

ФИО+ДР+СНИЛС/СНИЛС = 1_2

- ФИО: 0

- СНИЛС: 0

Карточка Е

ФИО+ДР=1_1

- ФИО: 0

ФИО+ДР+СНИЛС/СНИЛС=1_2

- ФИО: 0

- СНИЛС: 0

ФИО+ДР=1_1

- ФИО: 1

ФИО+ДР+СНИЛС/СНИЛС=1_2

- ФИО: 1

- СНИЛС: 0

Карточка А

#	ФИО/Имя/0
0	Александр
1	Александр

#	СНИЛС/СНИЛС/1
0	233

Карточка В

#	ФИО/Имя/0
0	Александр

#	СНИЛС/СНИЛС/1
0	

Карточка С

#	ФИО/Имя/0
0	Александр
1	Александр

#	СНИЛС/СНИЛС/1
0	288

Карточка D

#	ФИО/Имя/0
0	Александр

#	СНИЛС/СНИЛС/1
0	288

Карточка Е

#	ФИО/Имя/0
0	Александр
1	Александр

#	СНИЛС/СНИЛС/1
0	288

Имя тега	Закодированные данные		ID
ФИО+ДР	Александр		2_1
ФИО+ДР=СНИЛС/СНИЛС	Александр	233	2_2
ФИО+ДР=СНИЛС/СНИЛС	Александр	288	2_3

Узел 2

Карточка А

ФИО+ДР = 2_1 - ФИО: 0
ФИО+ДР=СНИЛС/СНИЛС = 2_2 - ФИО: 0 - СНИЛС: 0
ФИО+ДР = 2_1 - ФИО: 1
ФИО+ДР=СНИЛС/СНИЛС = 2_2 - ФИО: 1 - СНИЛС: 0

Карточка В

ФИО+ДР = 2_1 - ФИО: 0

Карточка С

ФИО+ДР = 2_1 - ФИО: 0
ФИО+ДР=СНИЛС/СНИЛС = 2_3 - ФИО: 0 - СНИЛС: 0
ФИО+ДР = 2_1 - ФИО: 1
ФИО+ДР=СНИЛС/СНИЛС = 2_3 - ФИО: 1 - СНИЛС: 0

Карточка D

ФИО+ДР = 2_1 - ФИО: 0
ФИО+ДР=СНИЛС/СНИЛС = 2_3 - ФИО: 0 - СНИЛС: 0

Карточка Е

ФИО+ДР = 2_1 - ФИО: 0
ФИО+ДР=СНИЛС/СНИЛС = 2_3 - ФИО: 0 - СНИЛС: 0
ФИО+ДР = 2_1 - ФИО: 1
ФИО+ДР=СНИЛС/СНИЛС = 2_3 - ФИО: 1 - СНИЛС: 0

Карточка А

ФИО+ДР = 3_1
- ФИО: 0

ФИО+ДР=СНИЛС/СНИЛС = 3_2
- ФИО: 0
- СНИЛС: 0

ФИО+ДР = 3_3
- ФИО: 1

ФИО+ДР=СНИЛС/СНИЛС = 3_4
- ФИО: 1
- СНИЛС: 0

Карточка С

ФИО+ДР = 3_3
- ФИО: 0

ФИО+ДР=СНИЛС/СНИЛС = 3_4
- ФИО: 0
- СНИЛС: 0

ФИО+ДР = 3_1
- ФИО: 1

ФИО+ДР=СНИЛС/СНИЛС = 3_2
- ФИО: 1
- СНИЛС: 0

Карточка Е

ФИО+ДР = 3_3
- ФИО: 0

ФИО+ДР=СНИЛС/СНИЛС = 3_4
- ФИО: 0
- СНИЛС: 0

ФИО+ДР = 3_1
- ФИО: 1

ФИО+ДР=СНИЛС/СНИЛС = 3_2
- ФИО: 1
- СНИЛС: 0

Карточка D

ФИО+ДР=3_1
- ФИО: 0

ФИО+ДР=СНИЛС/СНИЛС=3_2
- ФИО: 0
- СНИЛС: 0

Карточка В

ФИО+ДР = 3_3
- ФИО: 0

Узел 3

Имя тега	Закодированные данные		ID
ФИО+ДР	Борисович		3_1
ФИО+ДР=СНИЛС/СНИЛС	Борисович	445	3_2
ФИО+ДР			3_3
ФИО+ДР=СНИЛС/СНИЛС		445	3_4

Карточка А

#	ФИО/Отчество/0
0	Борисович
1	

#	СНИЛС/СНИЛС/2
0	445

Карточка В

#	ФИО/Отчество/0
0	

#	СНИЛС/СНИЛС/2
0	

Карточка С

#	ФИО/Отчество/0
0	
1	Борисович

#	СНИЛС/СНИЛС/2
0	445

Карточка D

#	ФИО/Отчество/0
0	Борисович

#	СНИЛС/СНИЛС/2
0	445

Карточка Е

#	ФИО/Отчество/0
0	
1	Борисович

#	СНИЛС/СНИЛС/2
0	445

Карточка А

ФИО+ДР = 4_1
- ДР: 0

ФИО+ДР+СНИЛС/СНИЛС = 4_2
- ДР: 0
- СНИЛС: 0

Карточка В

ФИО+ДР = 4_1
- ДР: 0

Карточка С

ФИО+ДР = 4_1
- ДР: 0

ФИО+ДР+СНИЛС/СНИЛС = 4_3
- ДР: 0
- СНИЛС: 0

Карточка D

ФИО+ДР = 4_1
- ДР: 0

ФИО+ДР+СНИЛС/СНИЛС = 4_3
- ДР: 0
- СНИЛС: 0

Карточка Е

Имя тега	Закодированные данные		ID
ФИО+ДР	15.03.1989		4_1
ФИО+ДР+СНИЛС/СНИЛС	15.03.1989	95	4_2
ФИО+ДР+СНИЛС/СНИЛС	15.03.1989	39	4_3

Узел 4

Карточка А

#	ДР/ДР/0
0	15.03.1989

#	СНИЛС/СНИЛС/3
0	95

Карточка В

#	ДР/ДР/0
0	15.03.1989

#	СНИЛС/СНИЛС/3
0	

Карточка С

#	ДР/ДР/0
0	15.03.1989

#	СНИЛС/СНИЛС/3
0	39

Карточка D

#	ДР/ДР/0
0	15.03.1989

#	СНИЛС/СНИЛС/3
0	39

Карточка Е

#	ДР/ДР/0
0	

#	СНИЛС/СНИЛС/3
0	39

Центр

Имя тега	Закодированные данные				ID
ФИО+ДР	1_1	2_1	3_1	4_1	1
ФИО+ДР+СНИЛС/СНИЛС	1_2	2_2	3_2	4_2	2
ФИО+ДР	1_1	2_1	3_3	4_1	3
ФИО+ДР+СНИЛС/СНИЛС	1_2	2_2	3_4	4_2	4
ФИО+ДР+СНИЛС/СНИЛС	1_2	2_3	3_4	4_3	5
ФИО+ДР+СНИЛС/СНИЛС	1_2	2_3	3_2	4_3	6

Карточка А

ФИО+ДР=1 - ФИО: 0 - ДР: 0
ФИО+ДР+СНИЛС/СНИЛС=2 - ФИО: 0 - ДР: 0 - СНИЛС: 0
ФИО+ДР=3 - ФИО: 1 - ДР: 0
ФИО+ДР+СНИЛС/СНИЛС=4 - ФИО: 1 - ДР: 0 - СНИЛС: 0

Карточка В

ФИО+ДР=3 - ФИО: 0 - ДР: 0

Карточка С

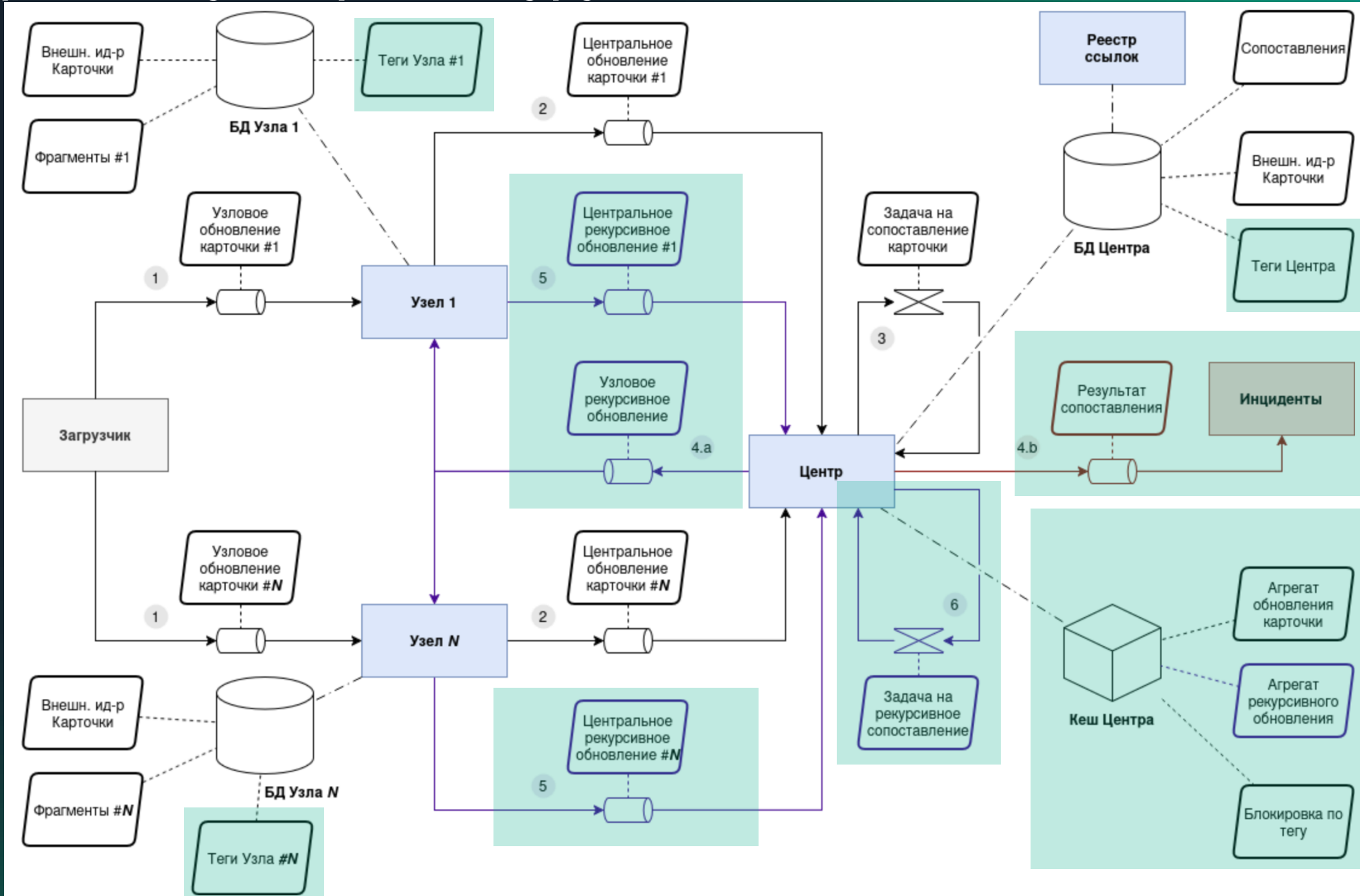
ФИО+ДР=3 - ФИО: 0 - ДР: 0
ФИО+ДР+СНИЛС/СНИЛС=5 - ФИО: 0 - ДР: 0 - СНИЛС: 0
ФИО+ДР=1 - ФИО: 1 - ДР: 0
ФИО+ДР+СНИЛС/СНИЛС=6 - ФИО: 1 - ДР: 0 - СНИЛС: 0

Карточка D

ФИО+ДР=1 - ФИО: 0 - ДР: 0
ФИО+ДР+СНИЛС/СНИЛС=6 - ФИО: 0 - ДР: 0 - СНИЛС: 0

Карточка Е

Совершенствуем архитектуру



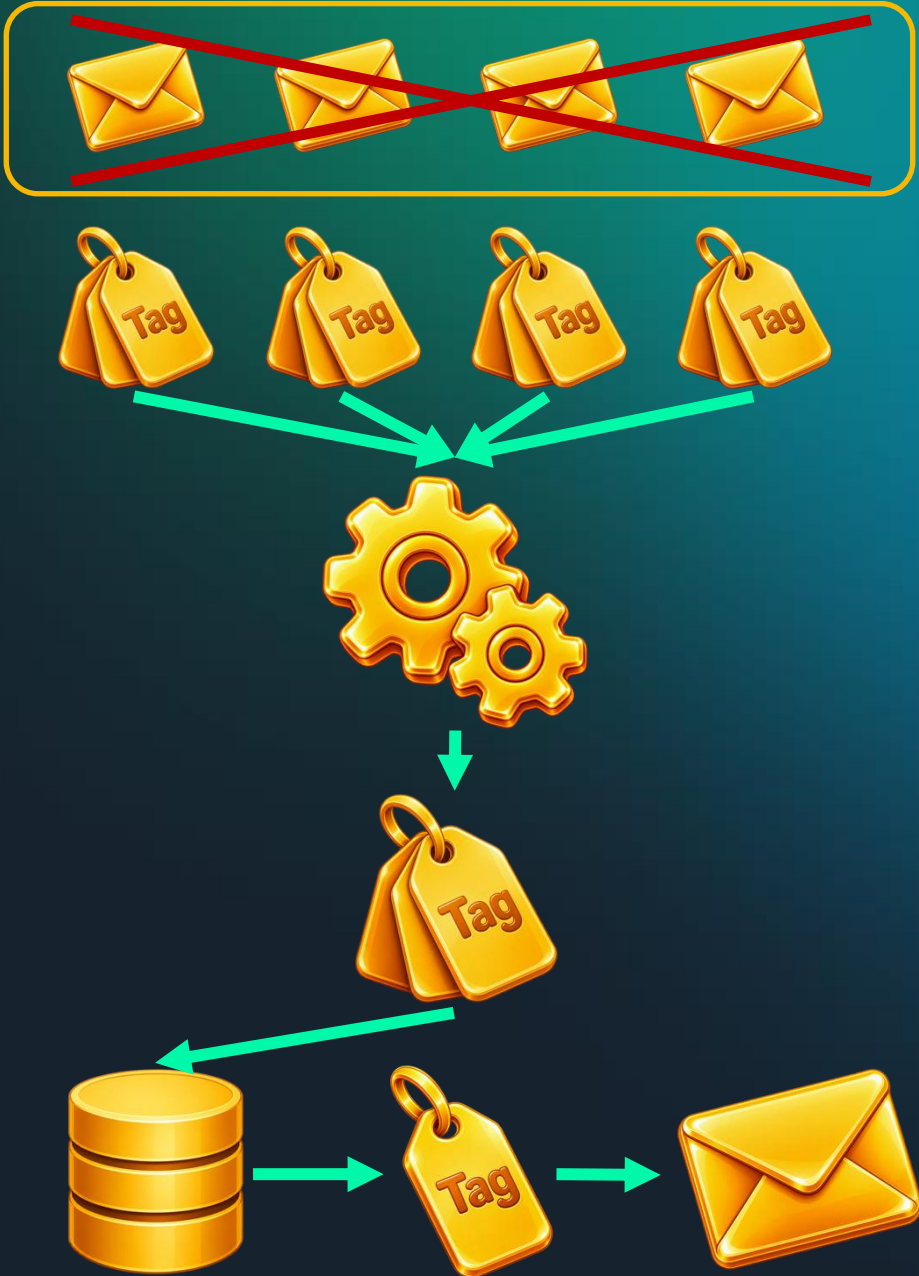
1. Узловое обновление

Имя	Тип	Описание
sourceId	String	Идентификатор источника Не может содержать пустое значение или значение «-»
sourceCardId	String	Идентификатор карточки в источнике.
version	Number	Версия обновления – монотонно возрастающая последовательность для истории изменений одной и той же карточки.
fragments	Array	Массив фрагментов, относящихся к данной карточке.
	• Object	Должен быть пуст при флаге deleted , установленном в true
	name	String Имя фрагмента. Например "ФИО/ФАМИЛИЯ/0".
	idx	Number Индекс значения свойства.
	val	String Значение фрагмента. Отсутствующее значение может быть представлено одним из следующих вариантов: • JSON-свойство val отсутствует • JSON-свойство val имеет значение null • JSON-свойство val равно пустой строке.
deleted	Boolean	Флаг, показывающий, удалась ли карточка.



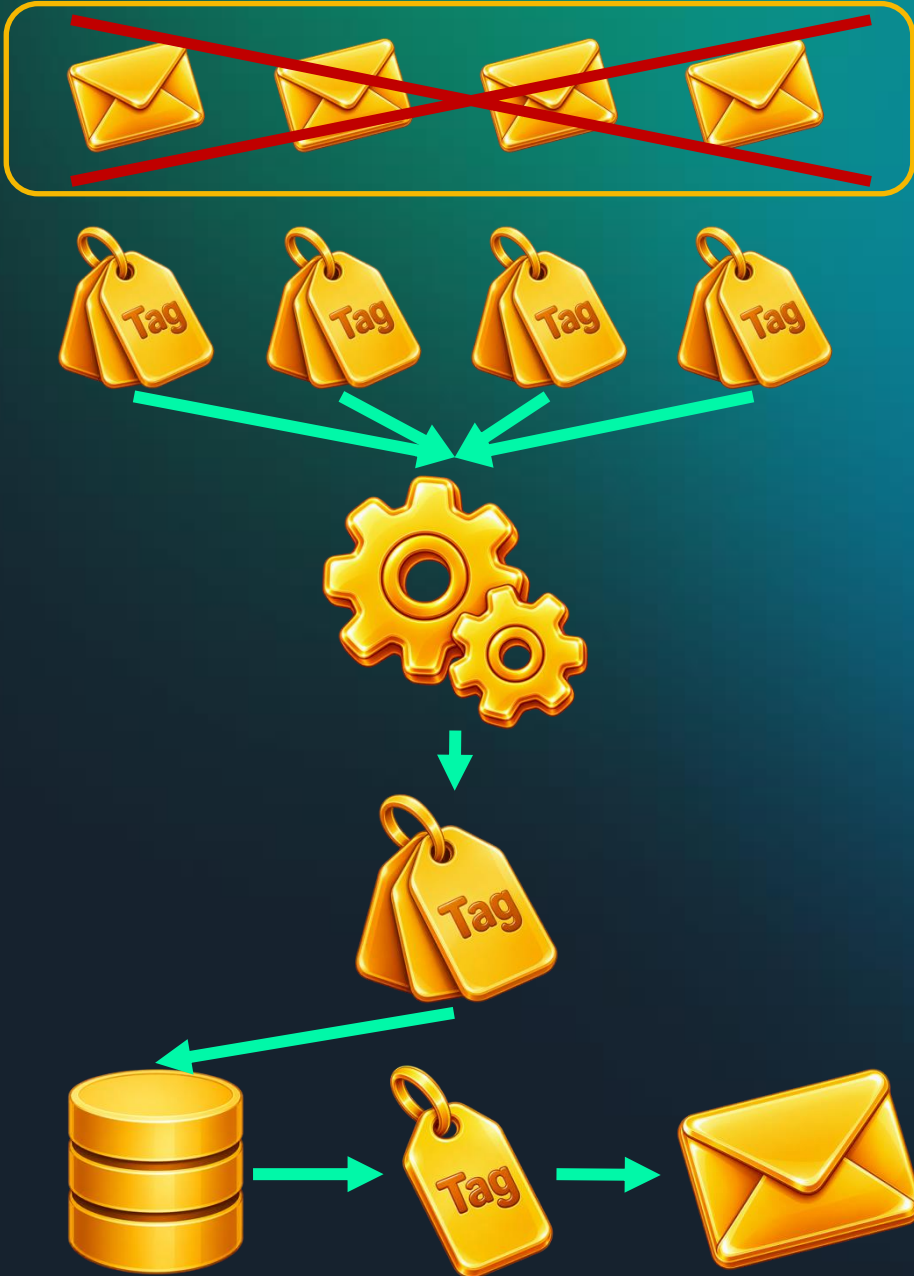
2. Центральное обновление

Имя свойства	Тип JSON	Описание
nodeId	String	Идентификатор Узла
sourceId	String	Идентификатор источника
cardId	String	Идентификатор исходной карточки в источнике
version	Number	Версия обновления – монотонно возрастающая последовательность для истории изменений одной и той же карточки
tags	Array Описание тега	Массив тегов Узла для данной карточки Пуст для удаляемой карточки, заполнен для новой и обновляемой карточки
deleted	Boolean	Флаг, показывающий, удалась ли карточка. Задается только при удалении исходной карточки



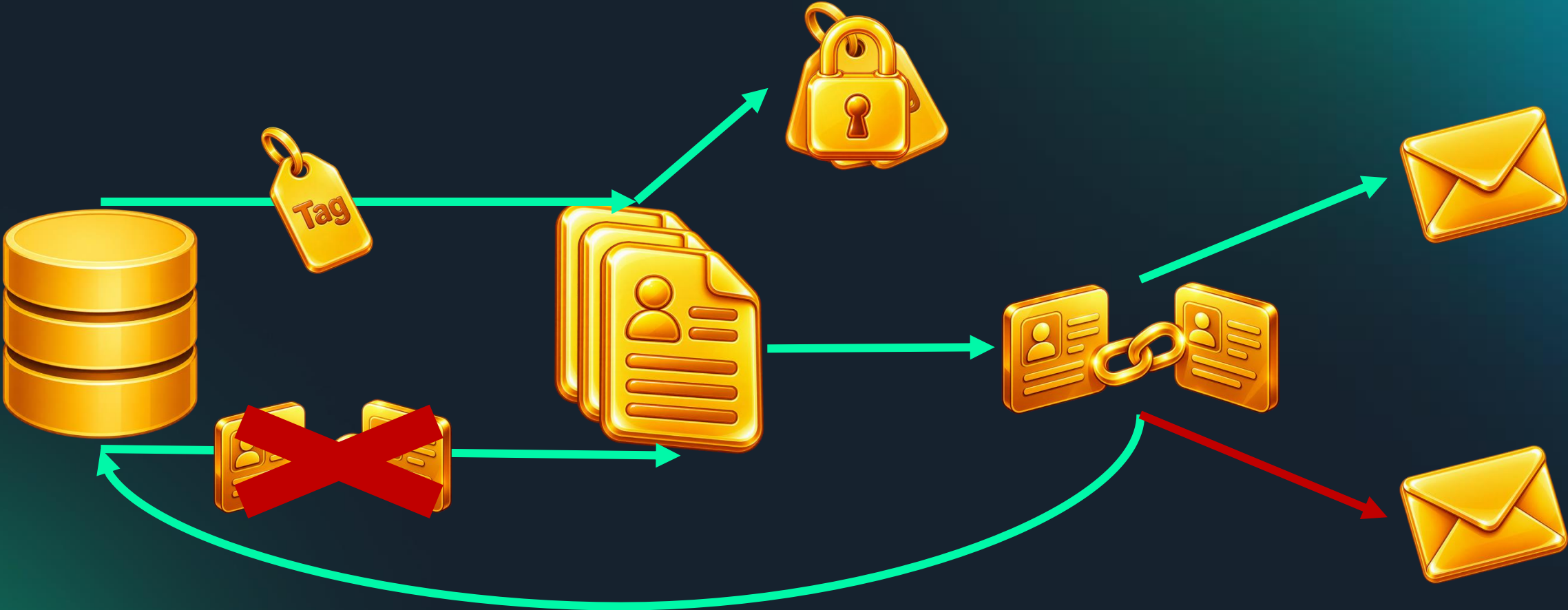
2. Центральное обновление

Имя свойства	
nodeId	
sourceId	
cardId	
version	
tags	1 той же
deleted	



3. Задача на сопоставление

Имя	Тип JSON	Описание
taskId	String	Идентификатор данной задачи на сопоставление
canonicTagId	Number	Идентификатор каноничного тега - стартовой точки для поиска кандидатов на сопоставление



Зачем нужны рекурсивные сопоставления

Пусть сконфигурированы следующие правила

Правило	ФИО	СОР	СНИЛС	Паспорт
ФИО+СОР+СНИЛС	REQ_EQUAL	REQ_EQUAL	REQ_EQUAL	OPT_CONSISTENT
ФИО+Паспорт	REQ_EQUAL	OPT_CONSISTENT	OPT_CONSISTENT	REQ_EQUAL

В систему поступили следующие карточки

Карточка/ сопоставление	ФИО	СОР	СНИЛС	Паспорт
Карточка А	Иванов А.Б.		112-288-445 39	5204 976841
Карточка В	Иванов А.Б.	VI-МЮ 134367		5204 976841
Карточка С	Иванов А.Б.	VI-МЮ 134367	112-288-445 39	

Карточки А и В сопоставляются по правилу ФИО+Паспорт, в результате заполнены как СОР, так и СНИЛС

Карточка/ сопоставление	ФИО	СОР	СНИЛС	Паспорт	Состоит из
Сопоставление #1	Иванов А.Б.	VI-МЮ 134367	112-288-445 39	5204 976841	А + В

Карточка С не сопоставима ни с одной другой карточкой

Карточка/ сопоставление	ФИО	СОР	СНИЛС	Паспорт	Состоит из
Сопоставление #2	Иванов А.Б.	VI-МЮ 134367	112-288-445 39		С

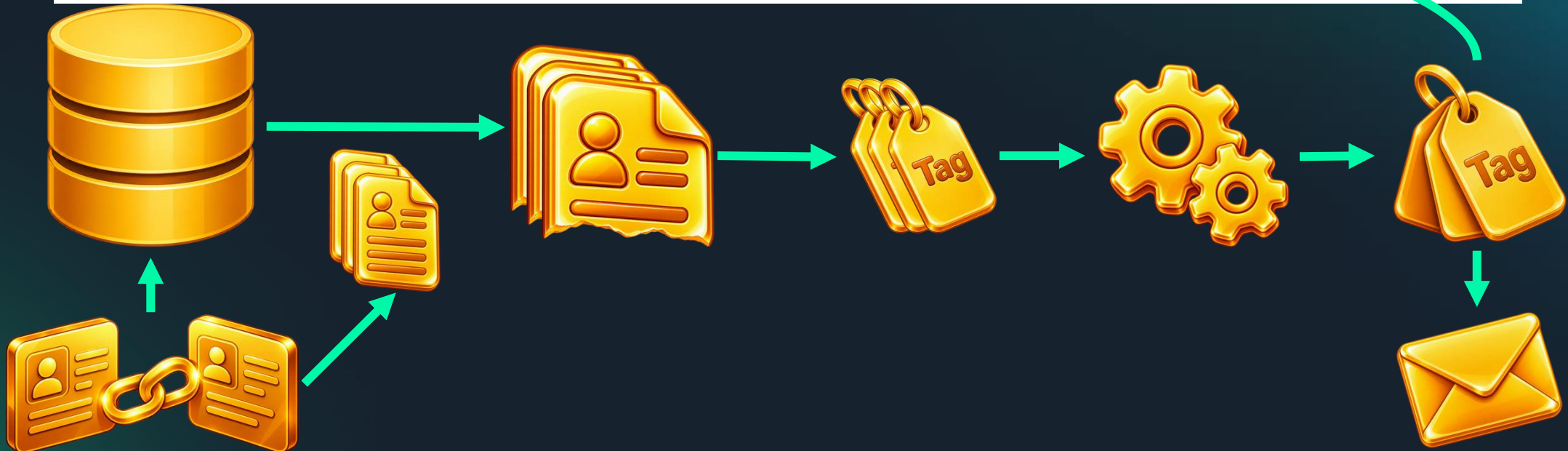
Благодаря рекурсивному сопоставлению, сопоставления #1 и #2 сопоставляются по правилу ФИО+СОР+СНИЛС

Карточка/ сопоставление	ФИО	СОР	СНИЛС	Паспорт	Состоит из
Сопоставление #3	Иванов А.Б.	VI-МЮ 134367	112-288-445 39	5204 976841	#1 + #2 = А + В + С

В результате, все три карточки А, В и С сопоставились между собой, что было бы невозможно без рекурсии

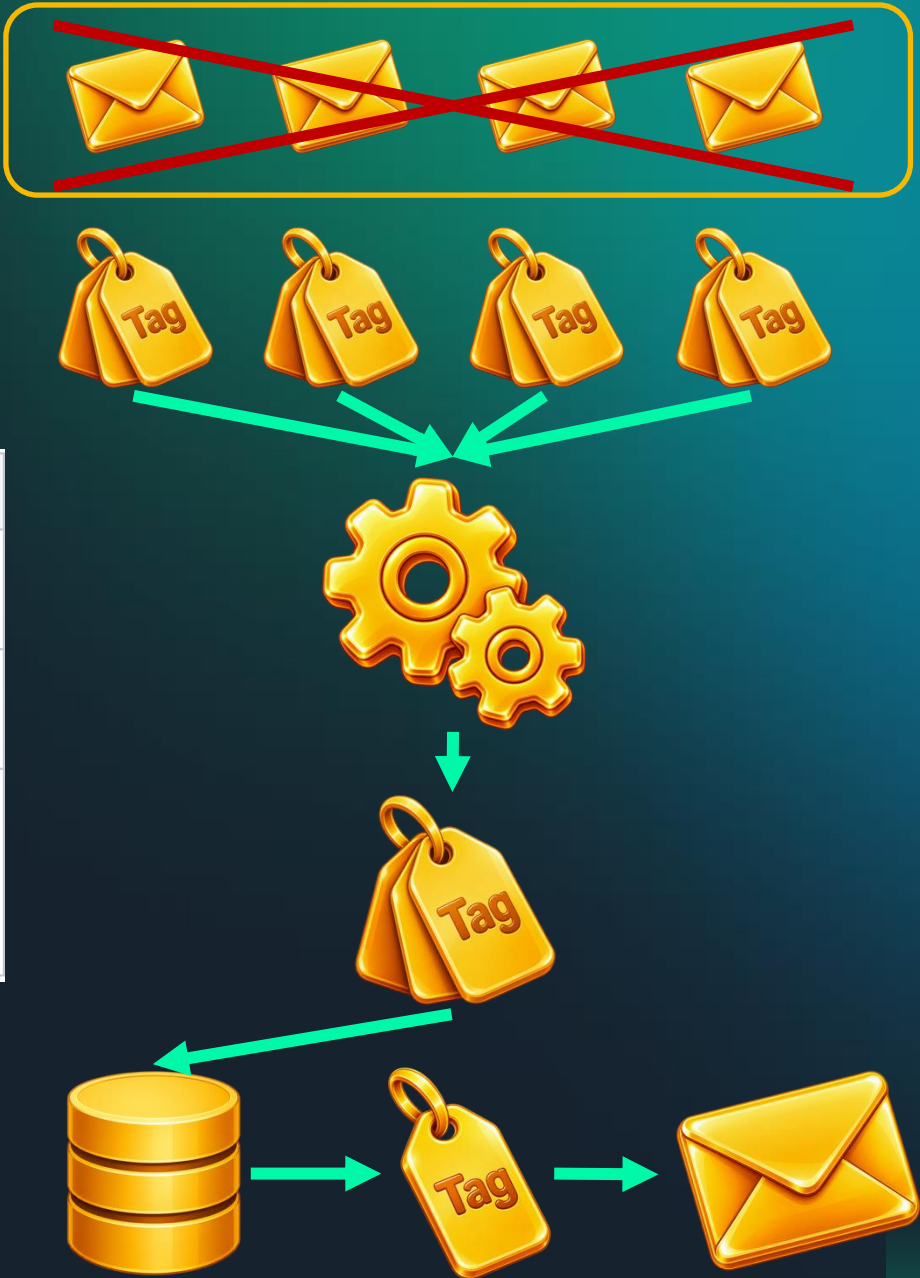
4.a Рекурсивное обновление Узла

Имя свойства		Тип	Описание
consolidationId		Number	Идентификатор добавляемой карточки сопоставления
cards		Array • Object	Множество исходных карточек, вошедших в добавляемую карточку сопоставления Передаются только исходные карточки, но не карточки сопоставления
	sourceId	String	Идентификатор источника.
	cardId	String	Идентификатор исходной карточки в источнике
	version	Number	Версия карточки – монотонно возрастающая последовательность для истории изменений одной и той же карточки.



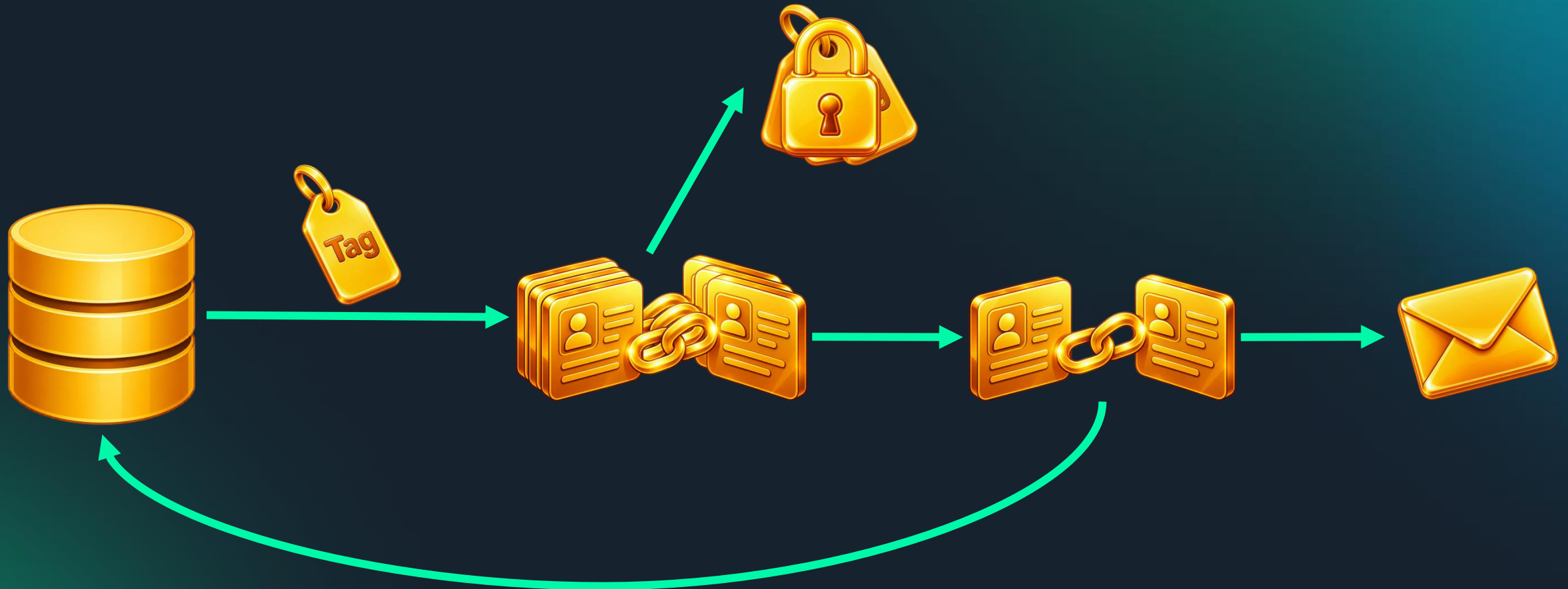
5. Рекурсивное обновление Центра

Имя свойства	Тип	Описание
nodeId	String	Идентификатор Узла
consolidationId	Number	Идентификатор сформированной карточки сопоставления
tags	Array Описание тега	Множество тегов Узла для сформированной карточки сопоставления



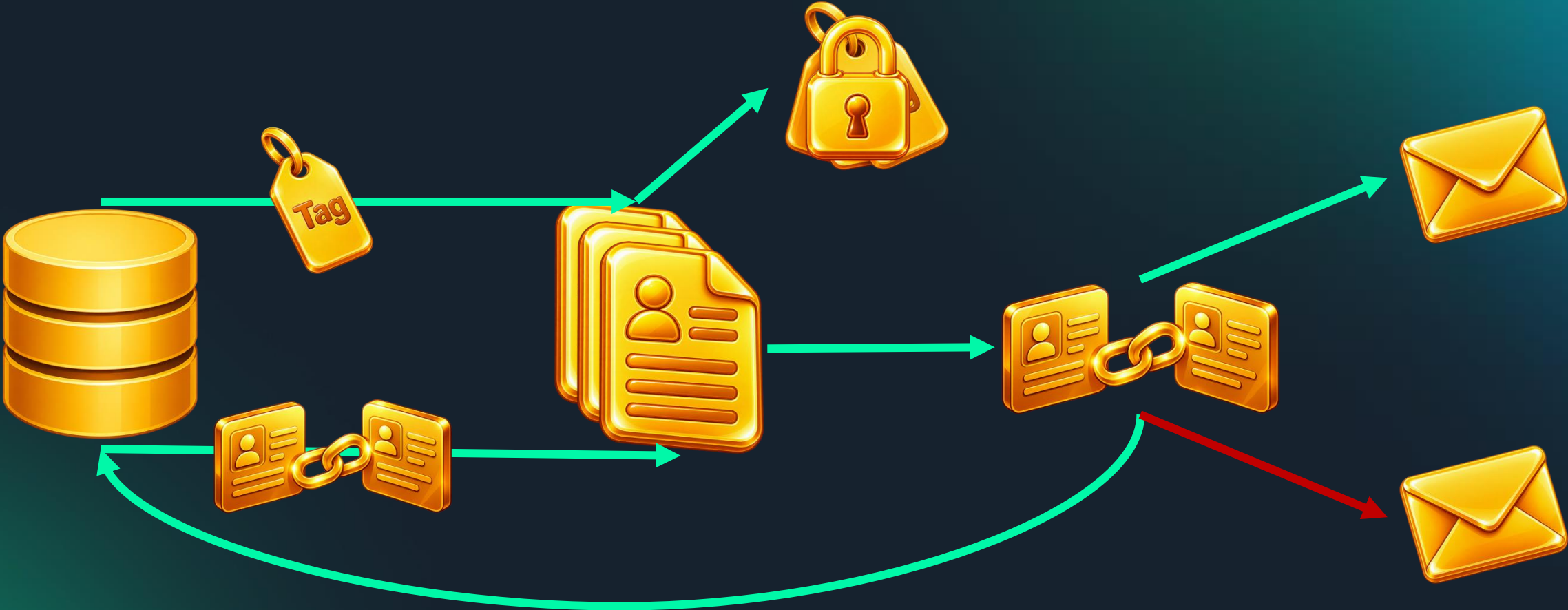
6. Задача на рекурсивное сопоставление

Имя	Тип JSON	Описание
<code>taskId</code>	String	Идентификатор данной задачи на сопоставление
<code>canonicTagId</code>	Number	Идентификатор канонического тега - стартовой точки для поиска кандидатов на сопоставление



3. Задача на сопоставление

Имя	Тип JSON	Описание
<code>taskId</code>	String	Идентификатор данной задачи на сопоставление
<code>canonicTagId</code>	Number	Идентификатор каноничного тега - стартовой точки для поиска кандидатов на сопоставление



4.b. Инциденты

Имя			Тип JSON	Обязательность	Описание
-			array object	Да	Массив выявленных групп сопоставления, сгруппированных по совпадающим значениям тега
	rule		string	Да	Идентификатор правила, по которому сопоставились данные карточки
	tag		number	Да	Значение тега, по которому осуществлялось сопоставление.
	consolidations		array - array - object	Да	Выявленные группы непротиворечивости в разрезе данного правила и значения тега
		sourceId	string	Да	Уникальный идентификатор Ведомственной витрины
		sourceCardId	string	Да	Внутренний идентификатор исходной карточки в Ведомственной витрине
		version	number	Да	Версия исходной карточки

- Карточка не сопоставилась
- Есть противоречия
- Есть дубли в источнике

ИТОГИ

Хороший разработчик может решить любую задачу.

А еще он может аргументированно обосновать, что задача неразрешимая, если ему лень эту задачу делать

- Решили задачу-оксюморон
- Учили все открытые вопросы v.1
- Обеспечили надежность через перепроигрывание карточек
- Скорость до 10 000 сопоставлений/сек

Давайте сопоставляться!



@goryntsev_y

