

CRM System for Telecom company

Report 2.

ПИ(англ)-2-23

Eldar Taiypov

Amirhan Ordobaev

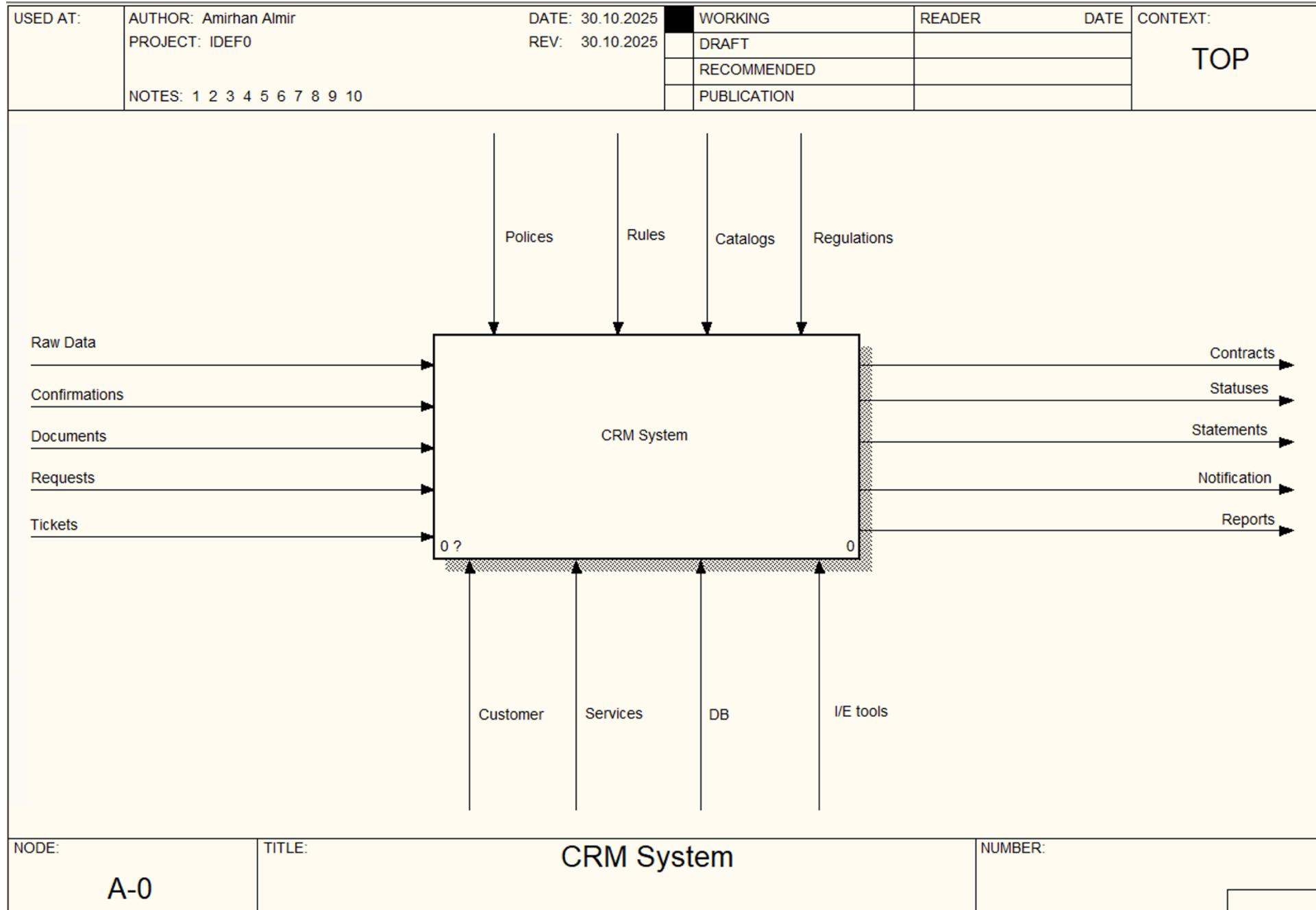
Almir Turdahunov

A-0 (контекст) — «CRM System»

Назначение. Показать границы системы и внешние интерфейсы.

- **Inputs (слева):** *Raw Data, Confirmations, Documents, Requests, Tickets* — входящие данные и события от клиентов/операторов/провайдера.
- **Controls (сверху):** *Polices* (политики/регламенты), *Rules* (правила валидации/билинга), *Catalogs* (каталог тарифов/опций), *Regulations* (нормы).
- **Outputs (справа):** *Contracts, Statuses, Statements, Notification, Reports* — создаваемые системой документы, статусы и уведомления.
- **Mechanisms (снизу):** *Customer* (персонал/пользователи), *Services* (внутр. сервисы, шедулер, биллинг), *DB* (БД), *I/E tools* (инструменты импорта/экспорта).

Результат работы системы на контексте: корректная обработка заявок/платежей, выпуск договоров и отчётов, своевременные уведомления.



A0 (верхний уровень) — «CRM System»

Состав верхнего уровня и ключевые интерфейсы между подсистемами:

1. Manage Customers (A1)

Формирует *List of Customers* и «кормит» A2 данными клиента.

2. Manage SIM and Tariffs (A2)

Принимает *Catalogs* как контроль; выдаёт *Accepted Contract* и *Statuses*.

3. Process Contracts and Activation (A3)

Принимает *Documents* и *Requests*; выпускает *Contracts*, *Statements*, *Activated Contract*.

4. Handle Payments and Balance (A4)

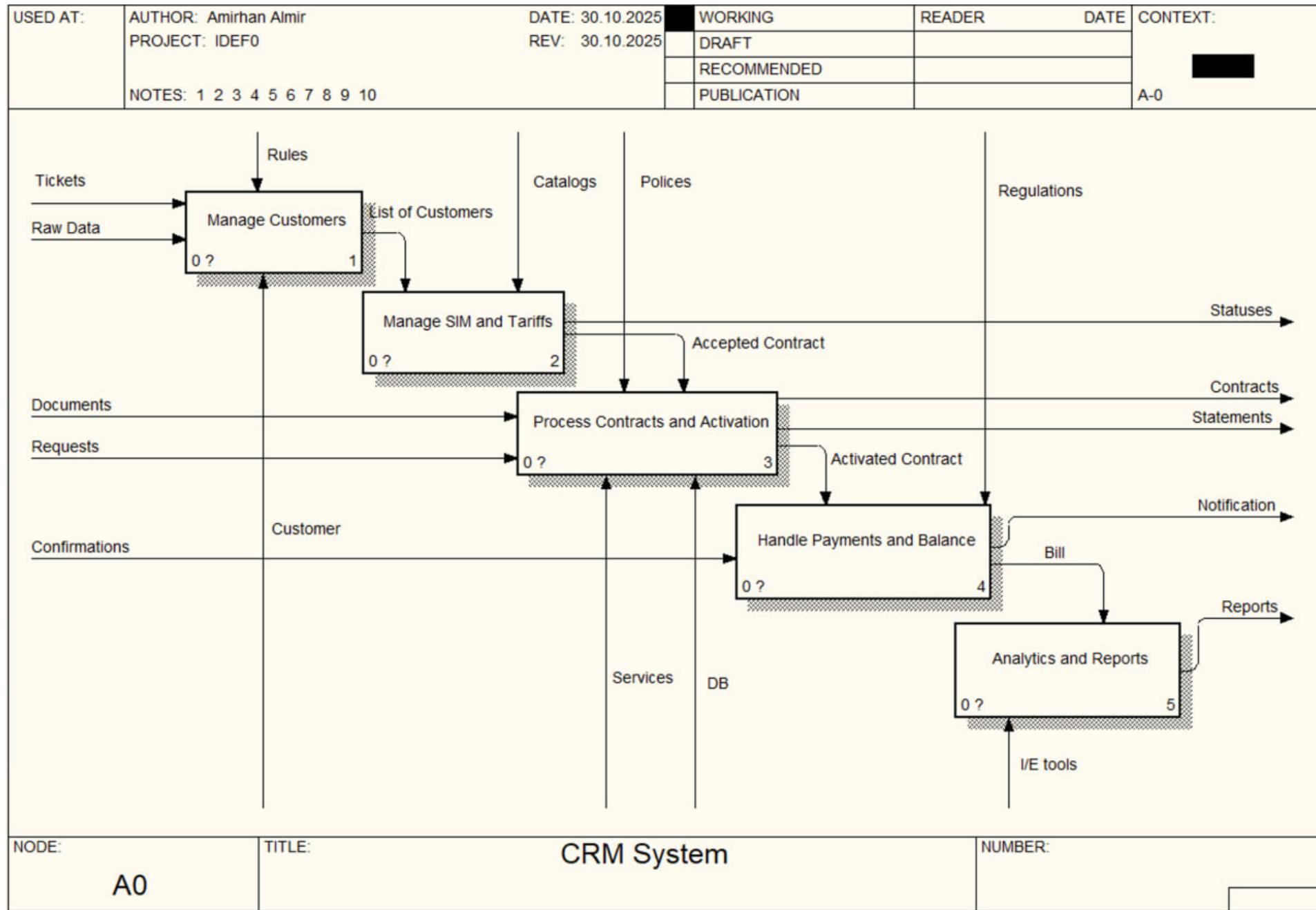
Принимает *Confirmations*; отправляет *Notification* и *Bill* (как отчёт/квитанцию).

5. Analytics and Reports (A5)

Выпуск *Reports* по данным предыдущих блоков.

Горизонтальные связи:

- *List of Customers* из A1 в A2.
- *Accepted Contract* из A2 — контроль для A3.
- *Activated Contract* из A3 — вход (control) в A4.
- *Bill* из A4 — контроль/источник для A5.

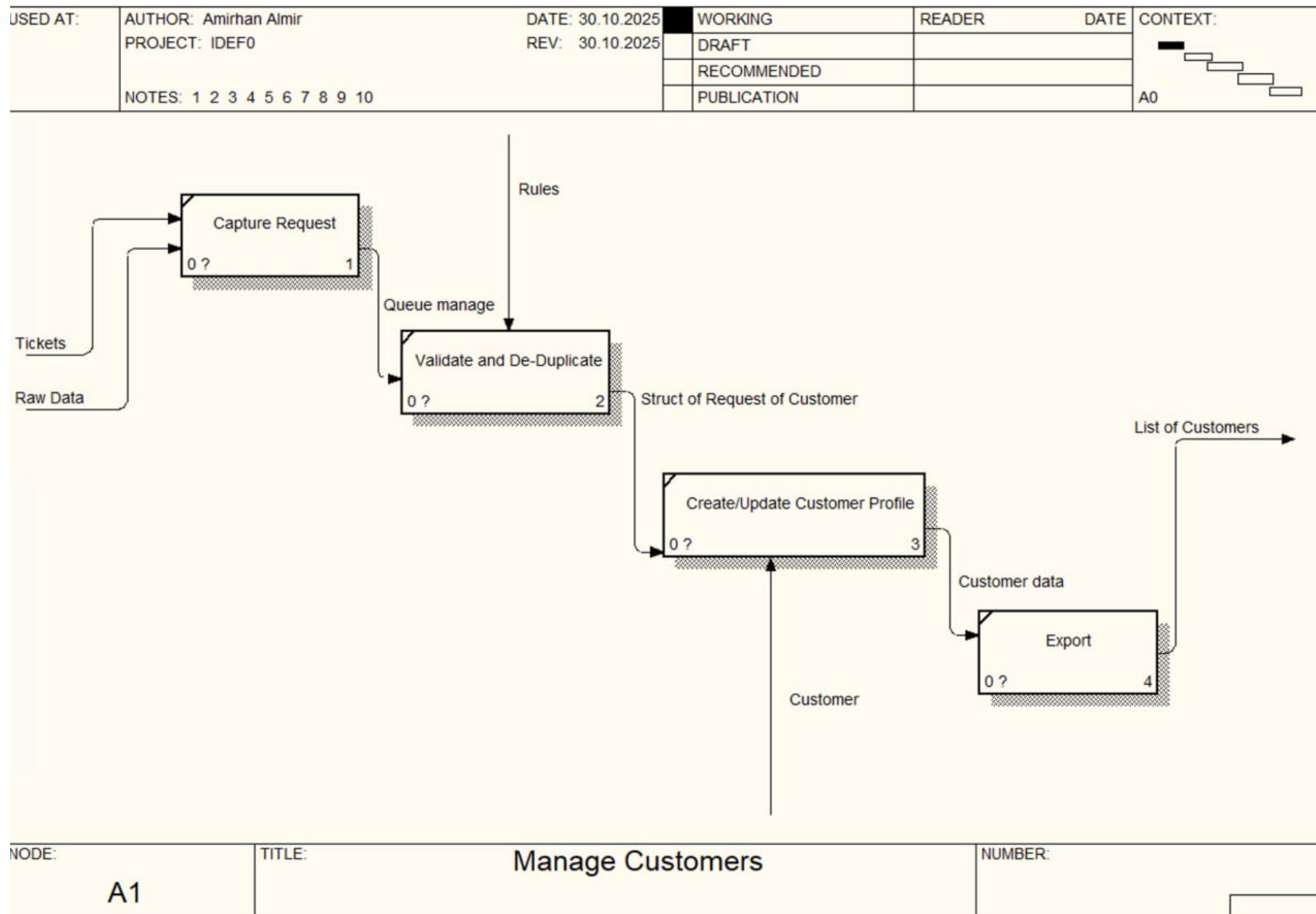


A1 — «Manage Customers»

Назначение. Ввод, проверка и ведение профилей клиентов; экспорт списков.

Функции:

- **A11 Capture Request** — приём заявки (*Tickets, Raw Data*).
 - Выдаёт внутренний поток *Queue manage*.
- **A12 Validate and De-Duplicate** — проверка по *Rules*, удаление дублей.
 - Формирует *Struct of Request of Customer*.
- **A13 Create/Update Customer Profile** — запись/обновление профиля (*Customer* как механизм).
 - Отдаёт *Customer data*.
- **A14 Export** — выгрузка профилей.
 - Внешний выход *List of Customers*.



A2 — «Manage SIM and Tariffs»

Назначение. Выбор клиента, привязка SIM и тарифа, выпуск черновика/принятого договора, обновление статуса.

Функции:

- **A21 Select Customer** — вход *List of Customers*; формирует *Customer fields*.
- **A22 Bind SIM and Tariff** — контроль *Catalogs*; на выход *SIM Intake*.
- **A23 Create Contract** — по данным клиента и SIM/тарифа выдаёт *Accepted Contract* и *Updated Contract* внутри.
- **A24 Notify and Update Status** — обновляет статусы, уведомляет; выход *Statuses*.

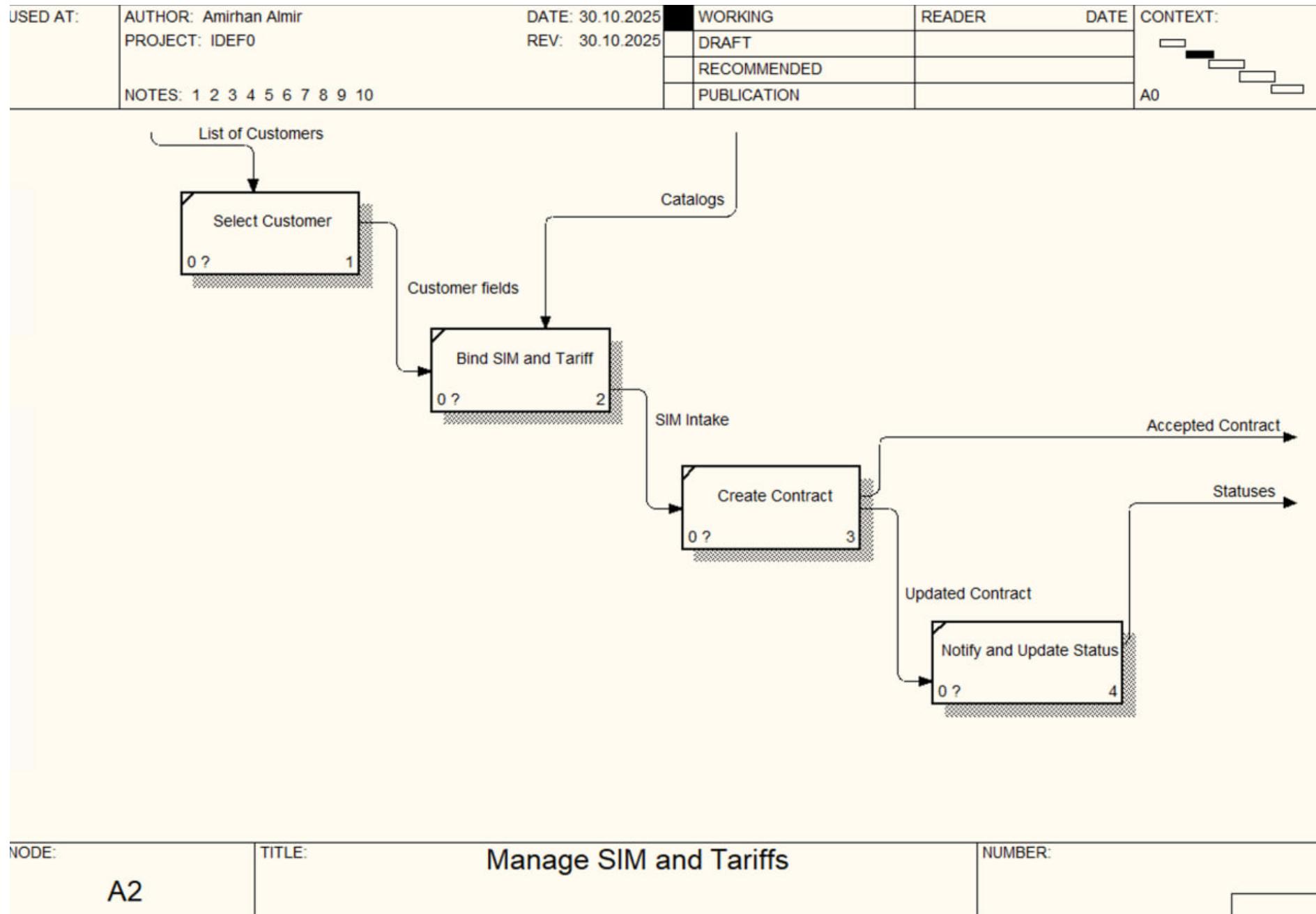
ICOM:

I: *List of Customers* (слева), *Customer fields*, *SIM Intake*.

C: *Catalogs*.

O: *Accepted Contract*, *Statuses*.

M: *Operator/DB/Services*.



A3 — «Process Contracts and Activation»

Назначение. Финализация условий, создание биллинговых политик, обновление заявки, активация.

Функции:

- **A31 Contract Schedule** — приём *Documents*; формирует *Contract with Documents intake*.
- **A32 Create Billing Polices** — контроль *Polices* и *Accepted Contract*; создаёт политики; выдаёт *Polices receipt*.
- **A33 Update Request** — вход *Requests* и *Polices receipt*; обновляет заявку (в БД) и формирует *Activation Summary*.
- **A34 Activate** — выполняет активацию по *Activation Summary*; выходы: *Contracts*, *Statements*, *Activated Contract*.

ICOM:

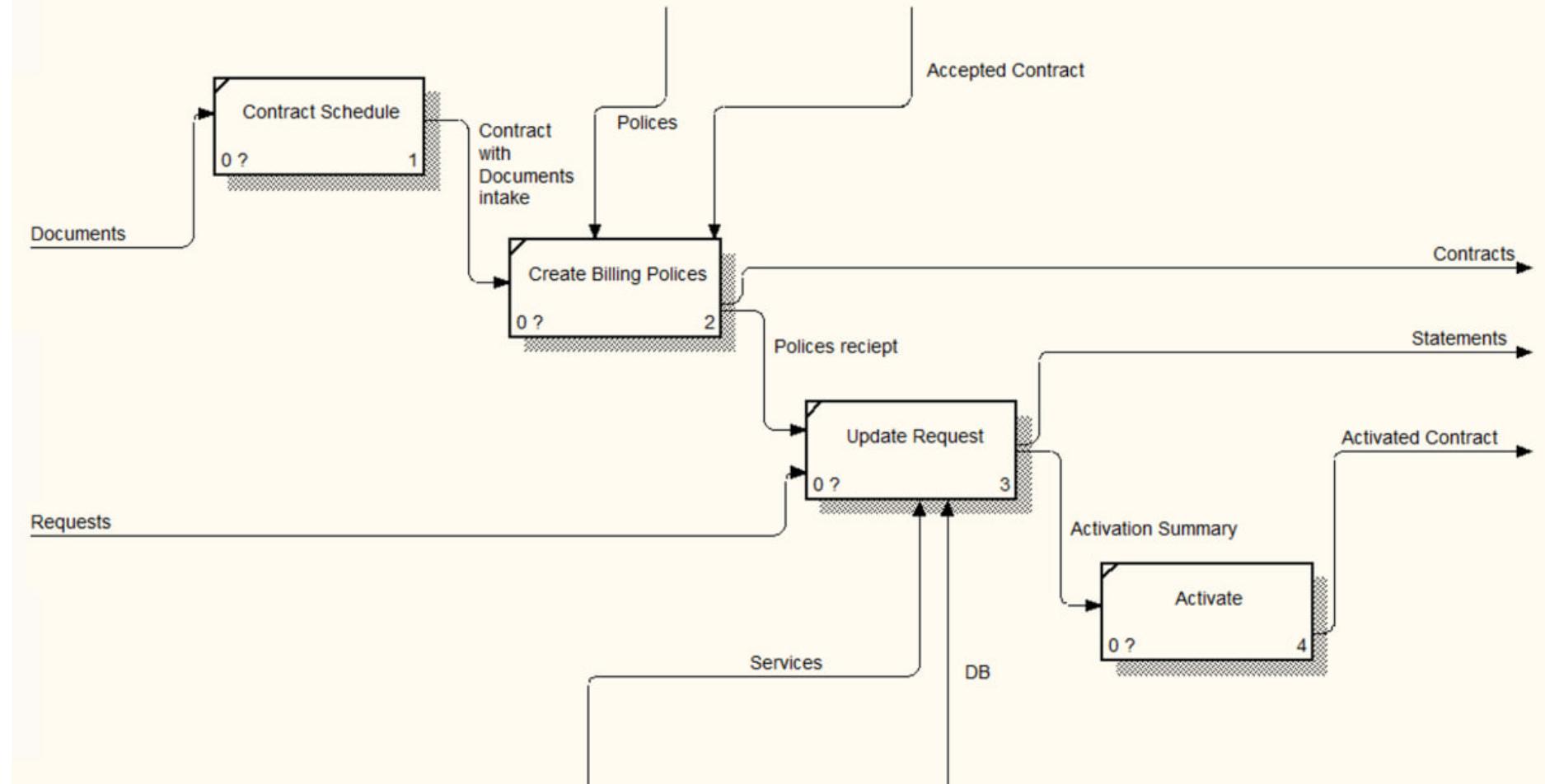
I: *Documents, Requests*.

C: *Polices* (регламенты), *Accepted Contract*.

O: *Contracts, Statements, Activated Contract*.

M: *Services, DB*.

USED AT:	AUTHOR: Amirhan Almir PROJECT: IDEF0	DATE: 30.10.2025 REV: 30.10.2025	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT:
	NOTES: 1 2 3 4 5 6 7 8 9 10					A0



NODE:	TITLE: Process Contracts and Activation	NUMBER:
A3		

A4 — «Handle Payments and Balance»

Назначение. Приём платежей, проведение и сверка, выпуск квитанций, нотификации.

Функции:

- **A41 Payment Intake and Validation** — приём платежа по *Activated Contract*; выдаёт *Payment Receipt*.
- **A42 Post Transaction update** — проводка/обновление баланса; использует *Confirmations*; передаёт запись дальше.
- **A43 Reconcile with Payment Gate** — контроль *Regulations*; сверка, формирование результата.
- **A44 Payment succesful** — завершение операции; выходы: *Notification, Bill*.

ICOM:

I: *Confirmations* (банк/провайдер).

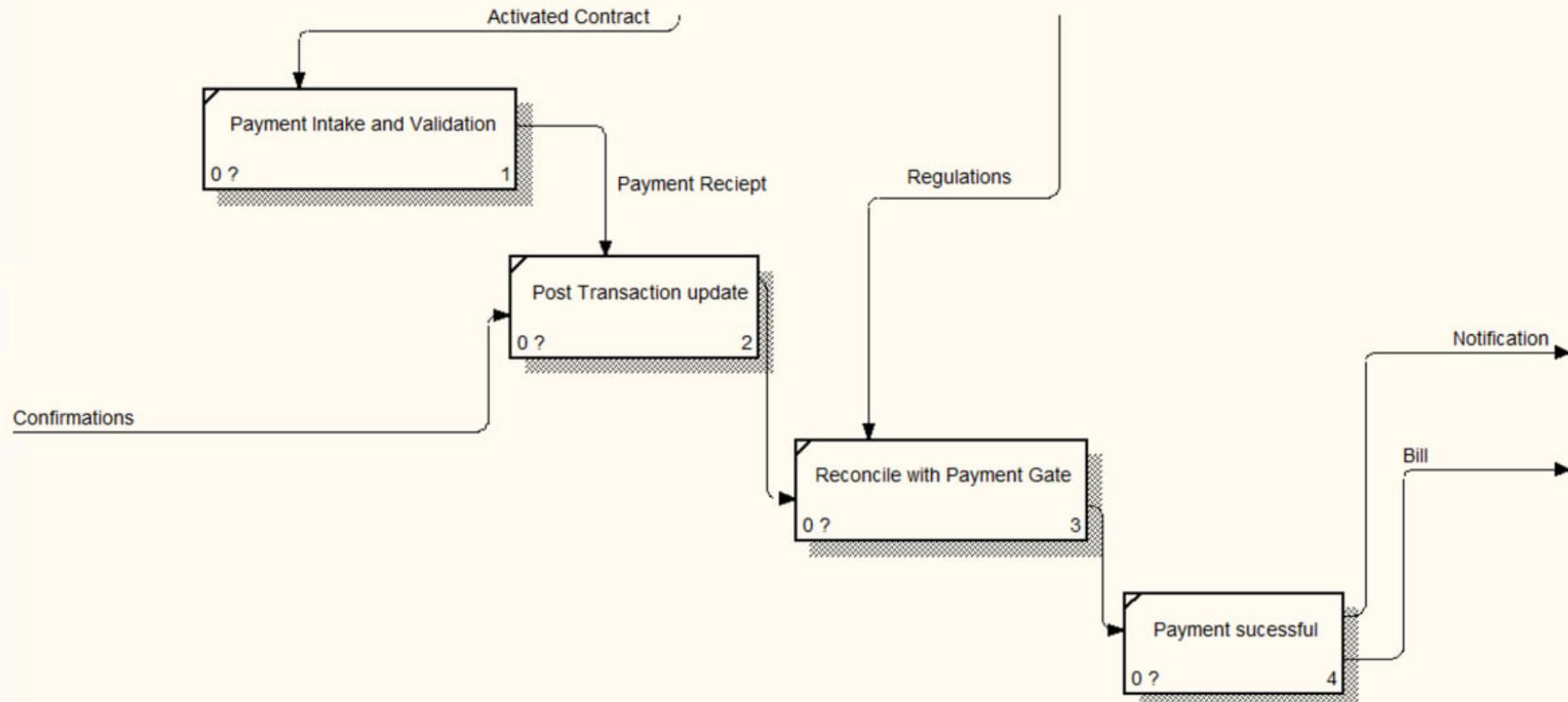
C: *Activated Contract, Regulations* (фин. нормы).

O: *Notification, Bill*.

M: *Payment gateway, DB, Email/SMS* (на листе обобщено как *Services*).

USED AT:	AUTHOR: Amirhan Almir PROJECT: IDEF0	DATE: 30.10.2025 REV: 30.10.2025	WORKING	READER	DATE	CONTEXT:
			DRAFT			
			RECOMMENDED			
			PUBLICATION			A0

NOTES: 1 2 3 4 5 6 7 8 9 10



NODE:	TITLE: Handle Payments and Balance	NUMBER:
A4		

A5 — «Analytics and Reports»

Назначение. Запросы на отчёты, вычисление метрик, ввести документ в силу.

Функции:

- **A51 Report request** — постановка требований к отчёту; даёт *Operational data* внутрь (фильтры/периоды).
- **A52 Compute Metrics** — расчёт показателей; отдаёт *Report metrics*.
- **A53 Publish Report** — выпуск и доставка; выход *Reports*.

ICOM:

I: *Operational data* (из A1–A4).

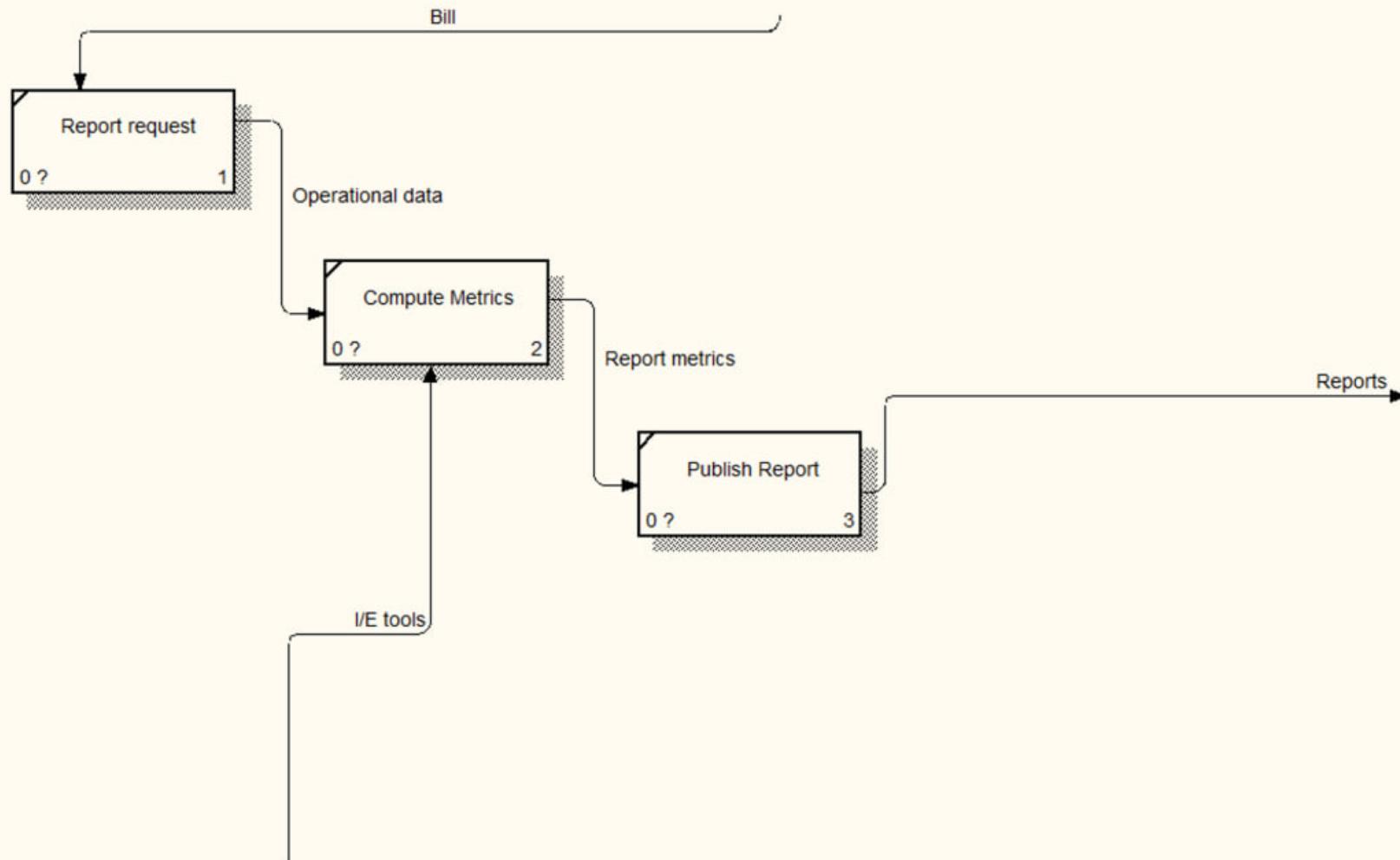
C: *Bill* (как контроль/источник расчётных периодов и шаблонов).

O: *Reports*.

M: *I/E tools* (ETL/BI), *DB*.

USED AT:	AUTHOR: Amirhan Almir PROJECT: IDEF0	DATE: 30.10.2025 REV: 30.10.2025	WORKING	READER	DATE	CONTEXT:
			DRAFT			
			RECOMMENDED			
			PUBLICATION			A0

NOTES: 1 2 3 4 5 6 7 8 9 10



NODE:	TITLE: Analytics and Reports	NUMBER:
A5		

DFD A-0 (Контекст) — «CRM System»

Цель уровня. Показать внешние акторы и обмен данными с системой без внутренних деталей.

Внешние сущности

- **Customer (E1).** Клиент/абонент оператора связи.
- **Admin (E2).** Администратор/оператор CRM, который просматривает и управляет данными.
- **PaymentGateway (E3).** Платёжный провайдер/банк.

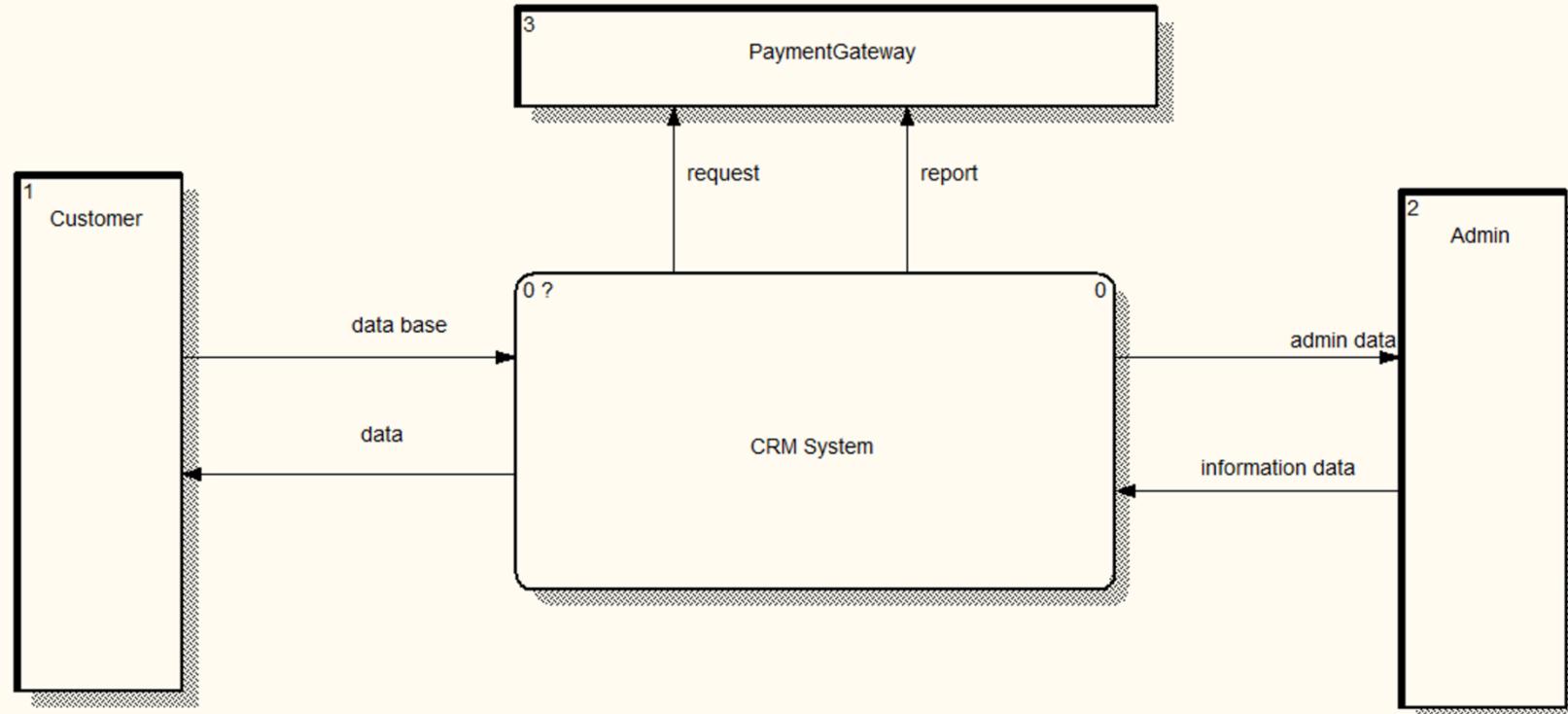
Процесс

- **P0: CRM System.** Единый контекстный процесс, агрегирующий всё поведение CRM.

Потоки данных

- От/к **Customer:**
 - data base → CRM — входные клиентские данные/анкеты (KYC), заявки. (*Рекомендуемое имя: Customer Data / Request*).
 - CRM → data — ответы/статусы, уведомления. (*Рекомендуемое имя: Status / Info*).
- Между **CRM** и **PaymentGateway**:
 - CRM → request — инициирование платёжных операций, авторизации.
 - report → CRM — подтверждения, отчёты/колбэки провайдера.
- От CRM к **Admin**:
 - admin data — административные данные для управления справочниками/доступами.
 - information data — оперативные сводки, отчётные данные, события.

USED AT:	AUTHOR: Eldar PROJECT: Operate_Telecom_CRM NOTES: 1 2 3 4 5 6 7 8 9 10	DATE: 30.10.2025 REV: 30.10.2025	WORKING	READER	DATE	CONTEXT: TOP
			DRAFT			
			RECOMMENDED			
			PUBLICATION			



NODE:	TITLE:	CRM System	NUMBER:
A-0			

DFD A0 (Level-1) — «CRM System»

Цель уровня. Показать основные подсистемы и их взаимодействие с хранилищами данных.

Процессы

- **P1: Customer_Management.** Принимает request от клиента, ведёт профили.
 - Хранилище: **D1 Customer_Data** — анкеты/профили клиентов.
 - Выход: Customer info → P2.
- **P2: SIM_Management.** Управляет SIM/номерами, их статусами/привязками.
 - Вход: Customer info от P1.
 - Хранилище: **D2 SIM_Data** — пул SIM и история состояний.
 - Выход: SIM Info → P3.
- **P3: Tariffs_and_Options.** Читает тарифы/опции, проверяет совместимость.
 - Хранилище: **D3 Tariff_Data** — тарифный каталог.
 - Выход: Tariff info → P4.
 - На схеме также показан общий вход data base (агрегированный доступ к БД).
- **P4: Contracts_and_Activation.** Формирует договор, готовит активацию.
 - Входы: SIM Info (от P2), Tariff info (от P3).
 - Хранилище: **D4 Contract_Data** — договоры/статусы.
 - Выходы: Contract → D4; Contract Info → P5.
- **P5: Billing_and_Payments.** Проводит платежи, формирует отчёт/инфопотоки.
 - Вход: Contract Info (от P4).
 - Хранилище: **D5 Payment_Data** — транзакции/балансы.
 - Взаимодействие с **PaymentGateway**: CRM отправляет request, получает report.
 - Выходы наружу: information data → Admin (оперативные сводки/квитанции).

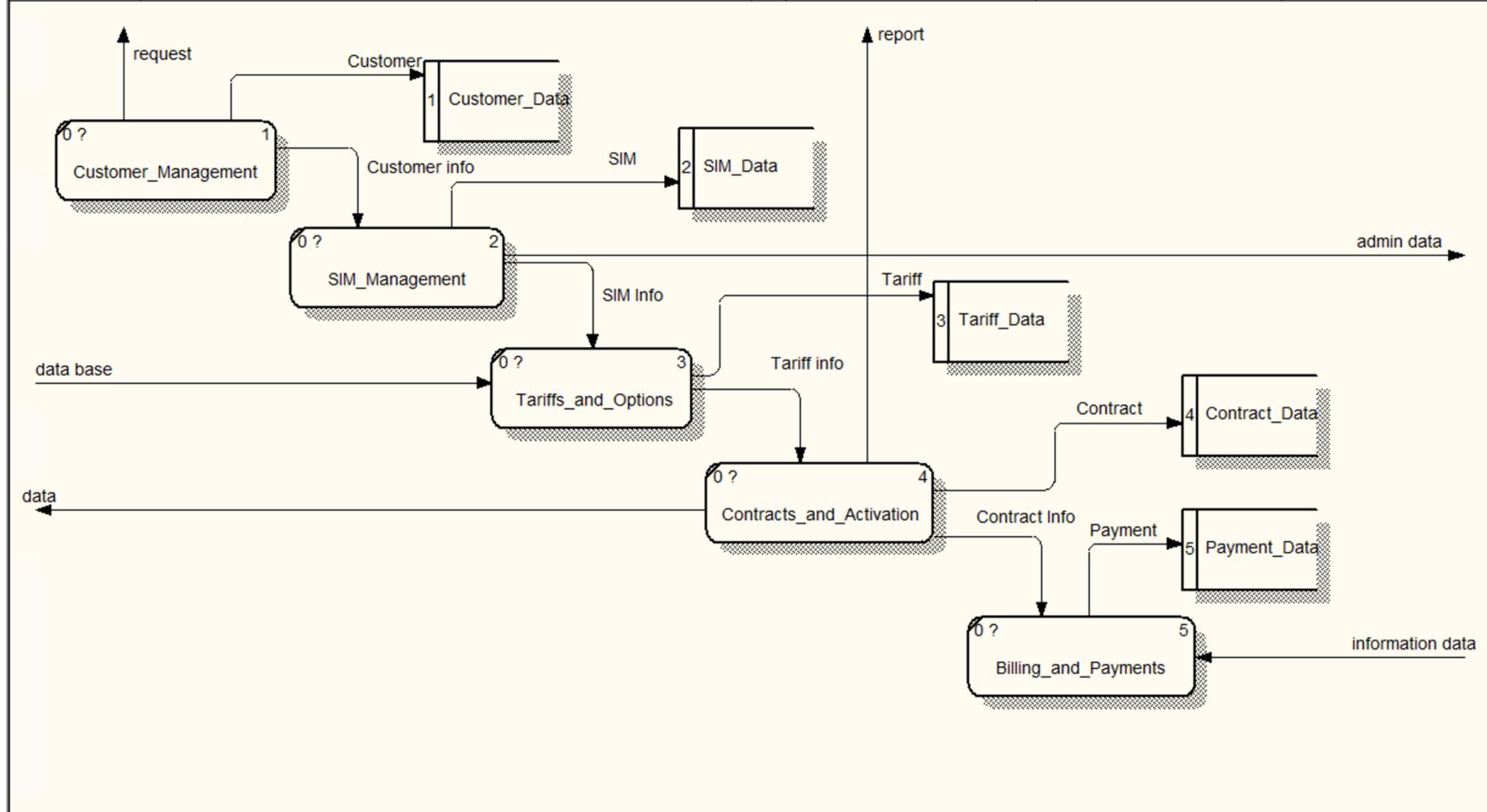
Хранилища данных

- **D1 Customer_Data** — профили/анкеты клиентов.
- **D2 SIM_Data** — идентификаторы/статусы SIM, история привязок.
- **D3 Tariff_Data** — тарифы и опции.
- **D4 Contract_Data** — договоры, статусы (active/suspended/closed).
- **D5 Payment_Data** — платежи, проводки, сальдо/балансы.

Внешние сущности (на уровне A0)

- **Customer** — источник request (заявок/действий).
- **Admin** — получатель information data (отчёты/оперативка), а также агрегированного admin data (на уровне A-0).
- **PaymentGateway** — принимает request, возвращает report (подтверждения).

USED AT:	AUTHOR: Eldar PROJECT: Operate_Telecom_CRM	DATE: 30.10.2025 REV: 30.10.2025	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT:
	NOTES: 1 2 3 4 5 6 7 8 9 10					A-0



NODE:	TITLE:	CRM System	NUMBER:
A0			

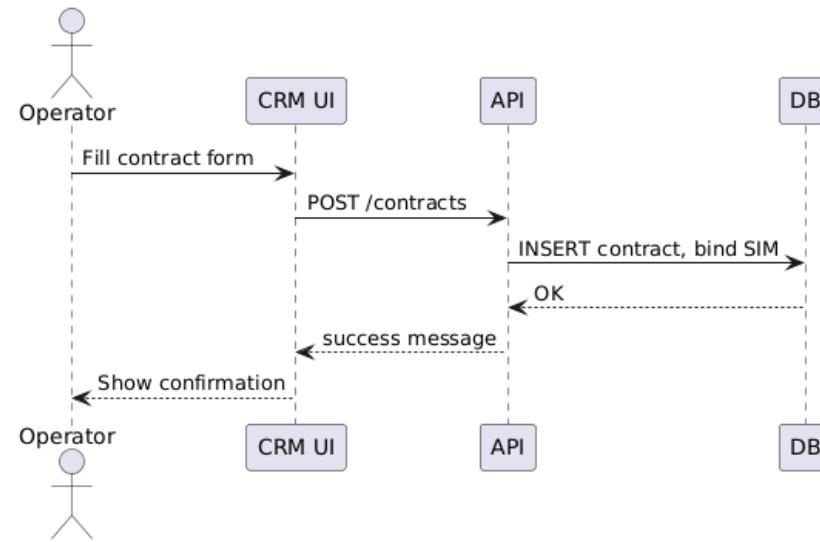
GetSim()

Общее описание

Диаграмма отображает процесс регистрации нового абонента и выдачи SIM-карты оператором связи через CRM-систему.

Участники:

- **Operator** — сотрудник компании, оформляющий договор;
- **CRM UI** — пользовательский интерфейс CRM-системы;
- **API** — серверная часть, принимающая и обрабатывающая запросы;
- **DB** — база данных, где хранятся сведения о клиентах и контрактах.



Описание процесса:

1. **Operator** заполняет форму договора в **CRM UI**.
2. Интерфейс отправляет запрос `POST /contracts` к **API**.
3. **API** обрабатывает запрос, создаёт новый контракт и выполняет операцию `INSERT contract, bind SIM` в **DB**.
4. После успешного добавления данных база возвращает ответ `OK`.
5. **API** передаёт сообщение об успешной регистрации обратно в **CRM UI**.
6. Оператор получает на экране подтверждение о выдаче SIM-карты клиенту.

Результат:

Абонент зарегистрирован, SIM-карта привязана к договору, а информация сохранена в базе данных.

GetDeposit()

Общее описание

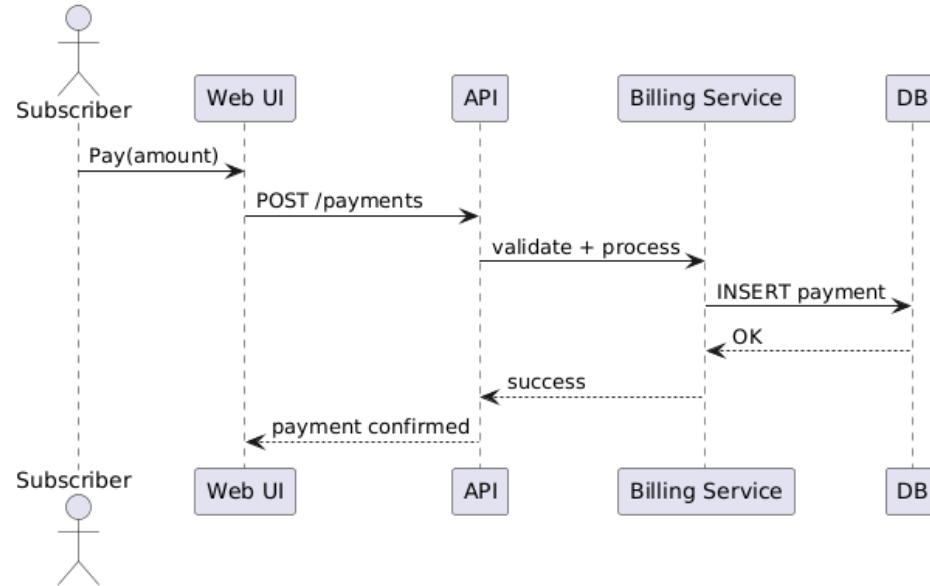


Диаграмма описывает взаимодействие абонента с биллинговой системой при оплате или пополнении счёта.

Участники:

- **Subscriber** — абонент, совершающий оплату;
- **Web UI** — веб-интерфейс самообслуживания;
- **API** — серверная часть, принимающая платёж;
- **Billing Service** — модуль биллинга, проверяющий и проводящий операции;
- **DB** — база данных платежей.

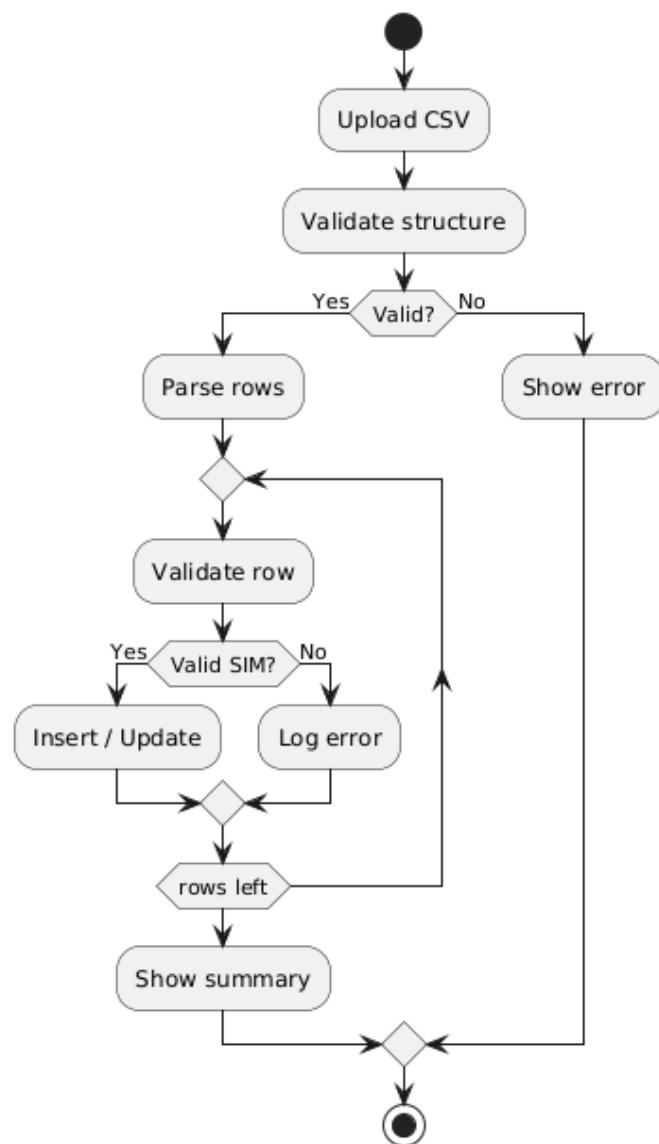
Описание процесса:

1. Абонент в **Web UI** вводит сумму и инициирует оплату (**Pay(amount)**).
2. Интерфейс отправляет запрос **POST /payments** к **API**.
3. **API** валидирует и передаёт данные в **Billing Service**.
4. Биллинговый модуль выполняет проверку и создаёт запись **INSERT payment** в **DB**.
5. После подтверждения база данных возвращает статус **OK**.
6. **Billing Service** уведомляет **API** об успешной оплате.
7. Абонент получает сообщение **payment confirmed** в интерфейсе.

Результат:

Платёж успешно обработан, сумма зачислена на счёт абонента, а запись о транзакции сохранена в биллинговой базе данных.

ManyRegister()



Общее описание

Диаграмма отражает процесс массовой загрузки и регистрации данных абонентов (например, корпоративных клиентов или SIM-пакетов) с помощью CSV-файла. **Этапы процесса:**

1. **Upload CSV** — оператор загружает CSV-файл с данными абонентов.
2. **Validate structure** — система проверяет структуру файла (корректность заголовков, формата данных).
 - Если структура **валидна**, выполняется переход к следующему шагу.
 - Если **ошибка**, система отображает *Show error*.
3. **Parse rows** — система построчно считывает данные.
4. **Validate row** — проверяется каждая запись: корректность номера, имени и SIM-идентификатора.
5. Если **SIM валидна**, выполняется *Insert / Update* в базу данных.
Если нет — фиксируется *Log error*.
6. Процесс повторяется, пока остаются необработанные строки.
7. После завершения формируется *Show summary* — итоговый отчёт с количеством успешно добавленных и ошибочных записей.

Результат:

Система массово регистрирует абонентов, автоматически добавляя или обновляя их данные в CRM, обеспечивая быстрое подключение большого числа клиентов (например, корпоративных SIM-пакетов).

TicketStatus()

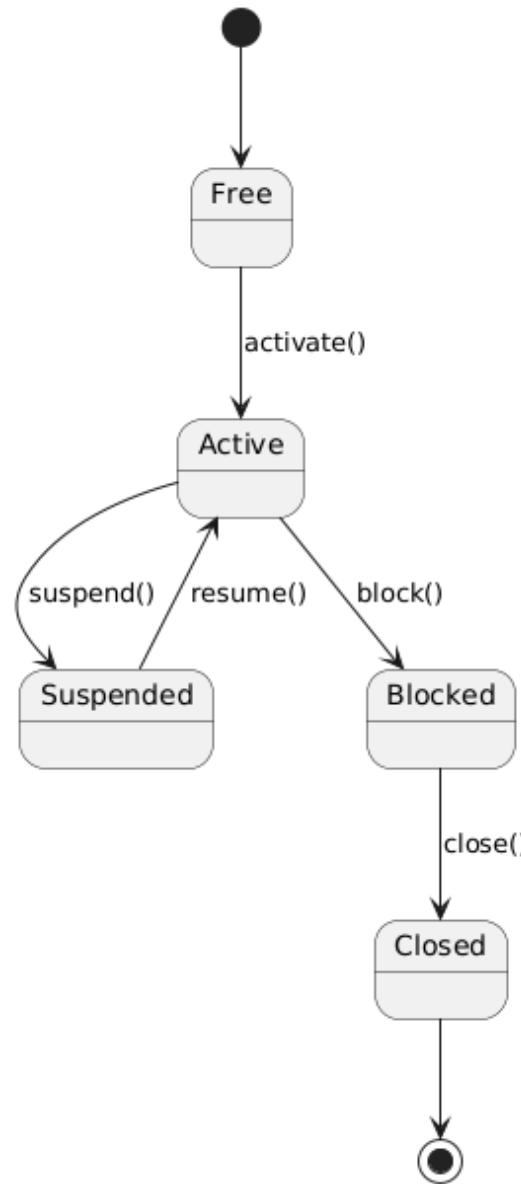
**Диаграмма состояний — TicketStatus()**

Диаграмма отражает жизненный цикл обращения (тикета) в системе технической поддержки.

Этапы процесса:

1. **Free** — исходное состояние тикета (неактивен).
2. **activate()** → **Active** — при активации тикет становится активным и готовым к обработке.
3. **Active** → **suspend()** → **Suspended** — при необходимости временно приостанавливается работа по тикету.
4. **Suspended** → **resume()** → **Active** — тикет возвращается в активное состояние.
5. **Active** → **block()** → **Blocked** — тикет может быть заблокирован, если требуется внешнее вмешательство.
6. **Blocked** → **close()** → **Closed** — после завершения всех действий тикет закрывается.

Результат:

Диаграмма демонстрирует все возможные состояния тикета и переходы между ними, что позволяет формализовать процесс его обработки и закрытия.

Package (Deployment) diagram

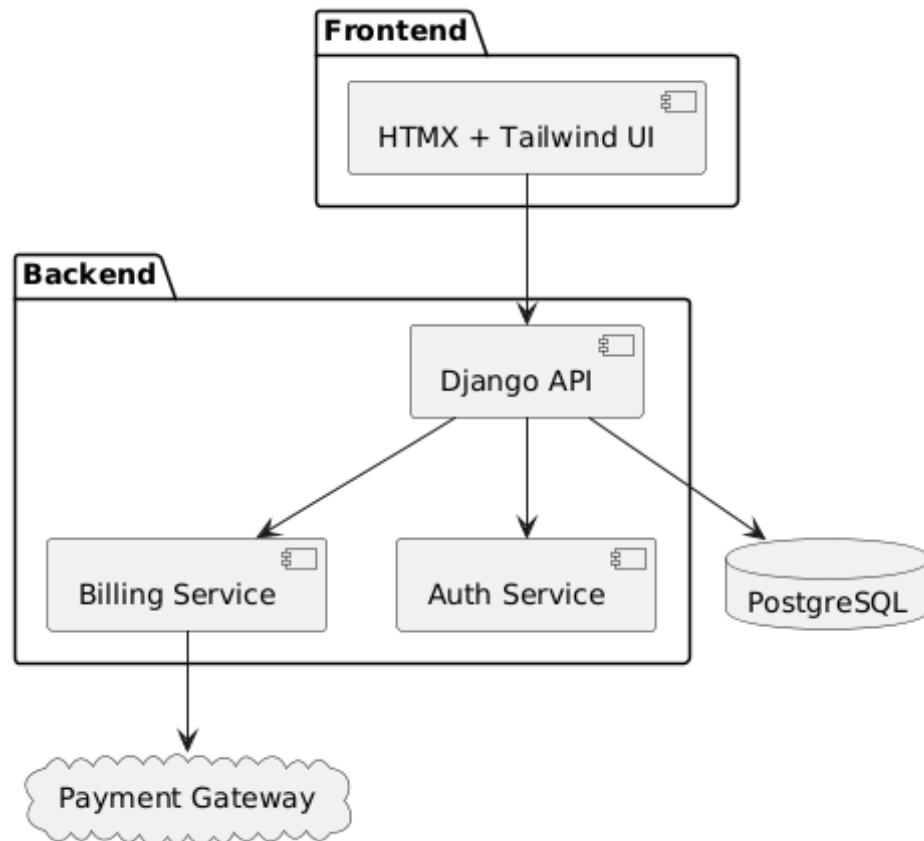


Диаграмма пакетов (Package / Deployment diagram)

Диаграмма описывает архитектурную структуру системы и взаимодействие её основных компонентов.

Компоненты:

1. **Frontend** — реализован с использованием *HTMX + Tailwind UI* для динамического интерфейса.
2. **Backend** — основан на *Django API*, который обеспечивает бизнес-логику и работу с данными.
3. **Billing Service** — отвечает за учёт и расчёт оплат.
4. **Auth Service** — обеспечивает аутентификацию и авторизацию пользователей.
5. **PostgreSQL** — база данных для хранения всей информации системы.
6. **Payment Gateway** — внешний сервис для проведения платежей.

Результат:

Диаграмма демонстрирует модульную структуру приложения, разделение фронтенда и бэкенда, а также интеграцию с внешними сервисами.

Class diagram

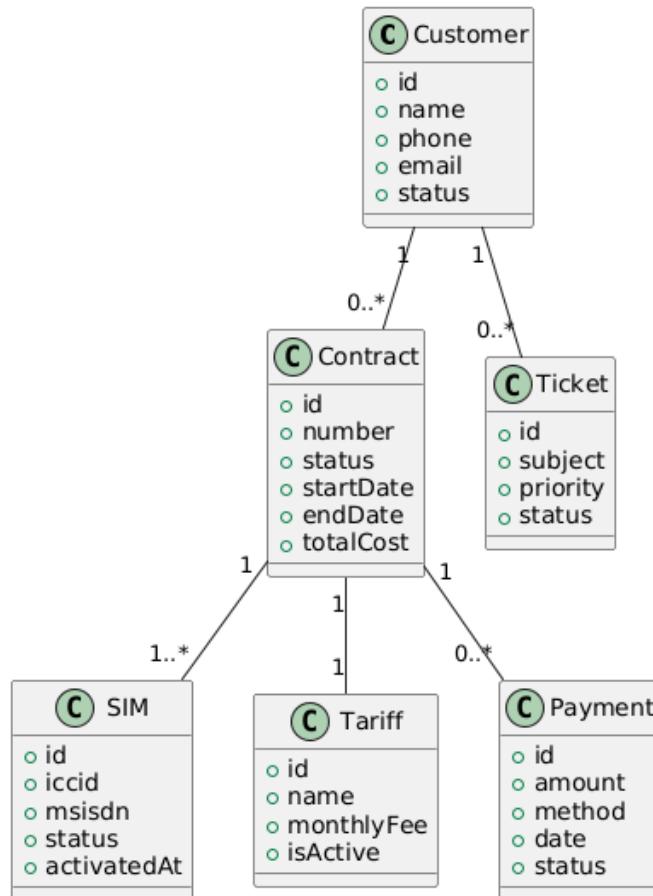


Диаграмма классов (Class diagram)

Диаграмма отражает структуру данных и взаимосвязи между основными сущностями системы.

Основные классы:

1. **Customer** — хранит информацию о клиенте (id, имя, телефон, email, статус).
2. **Contract** — представляет договор клиента, связывает *Customer* с *SIM*, *Tariff* и *Payment*.
3. **Ticket** — обращение клиента с атрибутами (тема, приоритет, статус).
4. **SIM** — сим-карта с уникальными параметрами (iccid, msisdn, статус, дата активации).
5. **Tariff** — тарифный план, включающий название, ежемесячную оплату и активность.
6. **Payment** — запись о платеже (сумма, метод, дата, статус).

Связи:

- Один *Customer* может иметь несколько *Contract*.
- Каждый *Contract* связан с одной *SIM* и одним *Tariff*.
- Один *Contract* может иметь несколько *Payment* и *Ticket*.

Результат:

Диаграмма определяет структуру данных и зависимости между сущностями, обеспечивая основу для реализации логики CRM-системы.

