



АНГАР PRO

АнгарПРО

TENT PROJECT

tentproject.ae

## KP Engine

# Архитектура KP Engine

Архитектура и операционная модель мультибрендового движка КП.

Документ Архитектура KP Engine	Собран 2026-03-30 01:24	Источник ARCHITECTURE.md
-----------------------------------	----------------------------	-----------------------------

<b>RU-first движок КП мультибренд / мультирынки / интеграции</b>	Документы собраны для продаж, внедрения и руководителя. Они показывают архитектуру, модель переменных, deployment и следующий слой функций без лишнего шума.
--	--

## Что это

KP Engine - это движок коммерческих предложений для тентовых и каркасно-тентовых решений. Он собран как RU-first продукт с нормальной адаптацией под UAE/GCC, а не как один HTML-файл с разными цветами.

Ключевой принцип:

- один core renderer;
- несколько брендов;
- несколько product families;
- один слой переменных;
- несколько способов выдачи: HTML, PDF, внешняя ссылка.

## Что держит систему

Слой	Что меняется	Что остается стабильным
brand	логотип, палитра, контакты, legal, кейсы, trust stats	renderer, секции и variable contract
product	тип конструкции, cover title, spec cards, estimate logic	бренд, регион и delivery
region	валюта, налог, локаль, климат, compliance copy	бренд и product logic
deal	клиент, параметры, цена, ROI, AI-тексты, менеджер	brand/product/region configs
template support	mapping исходных placeholders	core data model
integrations	amoCRM, Mango Office, Bitrix24 и другие adapters	клиентская верстка

Слой	Что меняется	Что остается стабильным
delivery	showcase, /kp/{deal_id}, PDF, public link	сама бизнес-логика КП

## Как проходит сделка

- 1 В систему приходит deal из CRM, webhook или ручной загрузки.
- 2 Выбираются brand, product, region.
- 3 Renderer собирает единый context.
- 4 Шаблон строит client-facing HTML.
- 5 Delivery слой отдает страницу, PDF и preview.
- 6 Интеграционный слой может вернуть статус обратно в CRM.

## Почему multi-brand здесь реальный, а не декоративный

Новый бренд подключается без форка renderer, если у него та же структура КП.

Типовой onboarding:

- 1 Добавить configs/brands/{brand\_id}.json.
- 2 Положить logo assets в static/logos.
- 3 Указать theme, contacts, legal, products, regions, cases.
- 4 Проверить /showcase, preview pages и PDF.

Что важно:

- brand config валидируется отдельным контрактом;
- docs и сами КП используют один и тот же brand source;
- RoyalTent в документации должен братья из brandbook asset в config, а не из отдельного "документного" логотипа;
- product и region слои переиспользуются, не копируются.

## Что уже работает

Контур	Состояние	Практический смысл
Multi-brand renderer	работает	один движок обслуживает RoyalTent, АнгарPRO и TentProject
Variable contract	работает	видно, какие поля прямые, какие вычисляются
HTML + PDF delivery	работает	КП можно открыть как ссылку или отдать файлом
Public showcase	работает	система показывается как продукт, а не как папка с файлами
amoCRM adapter	работает	сделка может стать источником runtime-данных

Контур	Состояние	Практический смысл
Mango Office adapter	частично	событийный слой есть, но не весь call lifecycle замкнут обратно в deal
Bitrix24 adapter	следующий шаг	точка подключения предусмотрена, runtime connector еще не закрыт

## Как на эту архитектуру садятся следующие функции

Следующий слой продукта не требует переписывать шаблон. Он садится поверх уже существующих границ:

- event adapters для CRM, телефонии и messaging;
- preprocessing services для разбора файлов, брифов и входящих материалов;
- analytics layer для просмотра КП, СТА и win/loss signal;
- approval layer для внутреннего согласования и клиентского подтверждения;
- asset workflow для 3D-рендеров, приложений и финальных PDF.

Именно поэтому roadmap можно расширять без второго движка и без копирования шаблонов под каждый бренд.

## Роль брендов в текущей системе

Бренд	Роль в системе
RoyalTent	премиальная client-facing линия, брендбук и презентационная подача
АнгарPRO	инженерная и industrial линия для ownership / ROI-сценариев
TentProject	UAE/GCC-ready линия для English-first и hospitality / event use-cases

## Что проверять при развитии системы

- бренд не должен вносить свою бизнес-логику в шаблон;
- новая интеграция не должна ломать variable contract;
- delivery не должен знать детали CRM;
- docs, showcase и client-facing pages должны брать одни и те же brand assets.

## Вывод

Эта архитектура сильна не тем, что "рендерит красивую страницу". Она сильна тем, что разделяет бренд, продукт, рынок, сделку и delivery. За счет этого продукт можно расширять на новые бренды, страны и каналы связи без переделки core-модели.