Техническое задание (ТЗ)

0. Карточка задачи (для портала Astana Hub)

Название: Миграция ПАК с Windows на NVIDIA Jetson Orin с переносимостью на Jetson Xavier и внедрением ML-оркестратора.
В офисе Заказчика
Цель: Запустить стабильный офлайн-конвейер (аудио/видео → извлекаемые признаки → агрегаторы → отчёт) на Orin с гарантированной переносимостью на Xavier.

Обязательное условие: Станции работают оффлайн (без доступа в интернет).

1. Цели и измеримые результаты (SLO / KPI)

1. Переносимость окружения: Один и тот же контейнерный стек запускается на Orin и Xavier (общая база на JetPack 5.1.x / L4T r35.x).

2. Оркестратор инференса: Предлагается
Вариант 1: NVIDIA Triton Inference Server (ensemble-графы) + связка с DeepStream 6.3 для видео и отдельным сервисом речи (Riva или faster-whisper).

Вариант 2: возможна реализация через NIVIDA JetPack, как рекомендованная сборка для разработки подобных задач)

3. Производительность (на AGX Xavier, 32 ГБ):

    - Видеоконвейер 720p@25–30 fps с отслеживанием лица и AU; rPPG окнами 20–30 с.

    - ASR стриминг kk/ru/en с задержкой до 600–900 ms на фразе 2–5 с.

    - LLM-диспетчер 3–7B (Q4) с откликом ≤1,5 s на короткую команду (≤64 токенов вывода).

4. Стабильность UI: Киоск-интерфейс (экран + озвучка) не просаживается при пиках инференса (FPS не ниже 24).

5. Отчёт: Формирование отчёта (ПАК→PDF) за ≤20 s после завершения сессии.

6. Трассируемость: В отчёте отображается техпаспорт расчётов (версии моделей/порогов/сборок).
7. Отчет формируется по утвержденному дизайну и с уже заданными полями и расчетами.

2. Объем работ (Scope)

2.1. Инвентаризация и план миграции

- Сбор текущего состояния (модели, скрипты, форматы данных).
- План соответствия JetPack 5.1.x: список контейнеров/базовых образов nvcr.io/nvidia/l4t-* r35.x; версии CUDA/TensorRT/DeepStream/Triton/Riva.
- Матрица совместимости Orin↔Xavier (AGX, NX), выделение «узких мест».

2.2. ТЗ по шифрованию

2.3. Оркестратор

- Triton Inference Server как единая точка сервинга (REST/gRPC, ensemble-графы)или иной вариант
- DeepStream 6.3 для видео: пайплайн детект → landmarks/AU → постпроцесс.

Или использование внутреннего ПО внутри ПАК в уже скомпилированном виде.

- Речь: сервис ASR/TTS (NVIDIA Riva или faster-whisper + Piper/Coqui TTS).
- LLM-роутер 3–7B (llama.cpp/Q4): форматирование E→A→I, function-calls.

2.4. Модели и форматы

- Храним ONNX/TorchScript; TensorRT .engine собираем на целевой машине (и кешируем по device-ID).
- INT8/FP16 профили; вынос лёгких INT8-сетей на NVDLA (где возможно).
- Файловая иерархия xxxcore/cache с версионированием.

2.5. UI/Audio изоляция

- Qt киоск в отдельном контейнере/неймспейсе, high-priority (chrt/taskset).
- GStreamer для воспроизведения стимулов (аппаратный декод).
- Поддержка USB-микромассивов с DSP (miniDSP UMA-8 / ReSpeaker XVF3800 / спикерфоны Poly/EPOS).

2.6. Логи, мониторинг и профилирование

- Системные: tegrastats, nvpmodel профили, термодатчики.
- Приложения: latency трекер по узлам (UI/ASR/видео/LLM/отчёт).
- Экспорт в локальные файлы; дашборд (lite) для техаудита.
- Обеспечить непрерывную запись данных в лог файлы для последующего анализа.

2.7. QC checking

- Вывод и сохранение логов в открытый системный файл.
- Формат сохранения в любом удобном варианте (md, txt, log)
- Запись в логах времени, кода ошибки, дополнительные примечания если они присутствуют самим ПО QC check

2.8. Документация и hand-over

- Manual: установка на Orin и на Xavier (пошагово).
- Cookbook: «как добавить новую модель в Triton», «как заменить ASR», «как выпускать обновления».
- CI: скрипты проверок (unit/интеграционные smoke-тесты).
- Запись вести в виде MarkDown (md) файлов

---

3. Нефункциональные требования

- Оффлайн: отсутствие зависимости от интернета (контейнеры и модели локально).
- Безопасность: минимум сетевых сервисов; локальные учётные данные; журналы доступов.
- Портативность: один и тот же compose-стек и конфиги для Orin и Xavier; различия — только в профилях TRT/DLA, собираемых на устройстве.
- Нагрузочная устойчивость: без троттлинга при nvpmodel -m 0; jetson_clocks; активное охлаждение.

- Explainable: лог вкладов моделей (для будущего отчёта E→A→I), версии и пороги.

---

4. Приемочные испытания (Acceptance)

4.1. Тестовая матрица устройств

- Jetson Orin (AGX / Orin NX) (16GB Mini).
- Jetson AGX Xavier (32 ГБ) и Xavier NX (8–16 ГБ). (опционально)

4.2. Сквозные сценарии

1. «Оффлайн-сессия 10 минут»: видео 720p + речь (kk/ru/en) + озвучка стимулов → отчёт ≤10 s.
2. «Пик нагрузки»: одновременный запуск видео-аналитики, ASR и TTS — UI FPS ≥25.
3. «Сборка на целевой»: при первом старте на новом устройстве автоматически собираются TensorRT engines, повторный старт — без пересборки.

4.3. Пороговые значения

- Видеоконвейер: ≥25 fps стабильно на 720p.
- ASR: средняя задержка ≤900 ms на фразу 2–5 s.
- LLM-роутер: ответ ≤1,5 s на запрос ≤64 токенов вывода.

4.4. Артефакты для приёмки

- Docker-образы, docker-compose.yml, конфиги DeepStream, репозиторий моделей Triton.
- Скрипты first-run.sh (сборка TRT) и collect-logs.sh.
- Инструкция (MD) по установке и эксплуатации.
- Видео-демо прохождения матрицы тестов (экраны/метрики).

---

5. Поставляемые материалы

1. Исходные коды пайплайнов, конфиги, Dockerfiles, compose и скрипты.
2. Репозиторий Triton с примерными моделями-заглушками (AU/детект/ASR/LLM).
3. Документация (установка, эксплуатация, добавление моделей).
4. Набор тестовых медиа и сценариев для приёмки.

---

6. Ограничения и правила разработки

- Не коммитить .engine из TensorRT; собирать на целевом устройстве.
- Fatbin для кастомных CUDA-плагинов с sm_72 (Xavier) и sm_87 (Orin).
- Жесткая фиксация версий образов: l4t-jetpack:r35.*, DeepStream 6.3 (опционально), совместимый Triton.
- UI/Audio — отдельные ядра CPU и повышенный приоритет планировщика.
- Запрещён исходящий интернет трафик.

---

7. Календарный план (согласовывается)

- Этап 1. Discovery & Plan: инвентаризация, план миграции, матрица совместимости.
- Этап 2. Контейнеризация: сборка образов, compose, базовый запуск на Orin.
- Этап 3. Оркестратор: Triton+DeepStream+ASR, репозиторий моделей, first-run build TRT.
- Этап 4. Тюнинг и приемка: SLO, матрица тестов, документация, hand-over.

---

8. Требования к подрядчику

- Опыт продакшн-проектов на Jetson (Orin/Xavier), DeepStream/Triton, оптимизация под TensorRT/INT8.
- Знание ASR/TTS (Riva/faster-whisper), GStreamer, Docker.
- Готовность передать ИС заказчику (код/образы/скрипты/документацию).

---

9. Условия и формат предложения

- Декомпозиция цены по этапам (фикс-прайс/вехи).
- Гарантийный период устранения дефектов (не менее 60 дней).
- Формат коммуникаций: еженедельный отчёт, доступ к репозиторию/CI.
