

voxlink



СБЕР
ЗДОРОВЬЕ

Архитектура VoIP

От одного Asterisk-а до гибрида

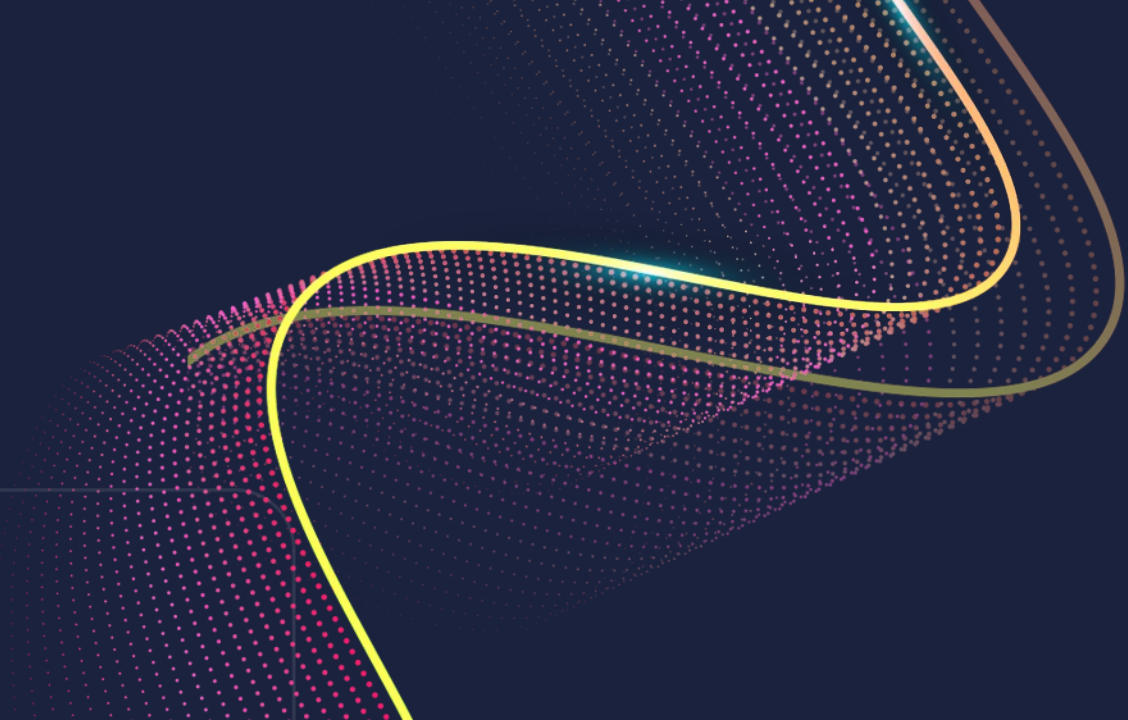
■ Учебный центр Voxlink



Артём Друзь

СберЗдоровье

- Образование — высшее (инженер ИТ)
- Инженер VoIP с опытом в ИТ сфере около 10 лет
- Участник различных VoIP-проектов в качестве внешнего специалиста
- Сертифицированный специалист OpenSIPS (OCP ID 7122-03)



01

Немного предыстории

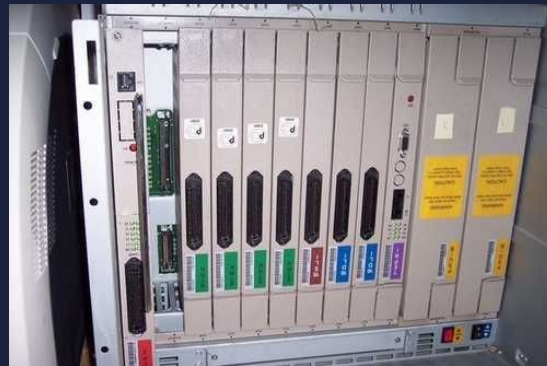
С чего начинались системы телефонии

Аналоговые АТСки



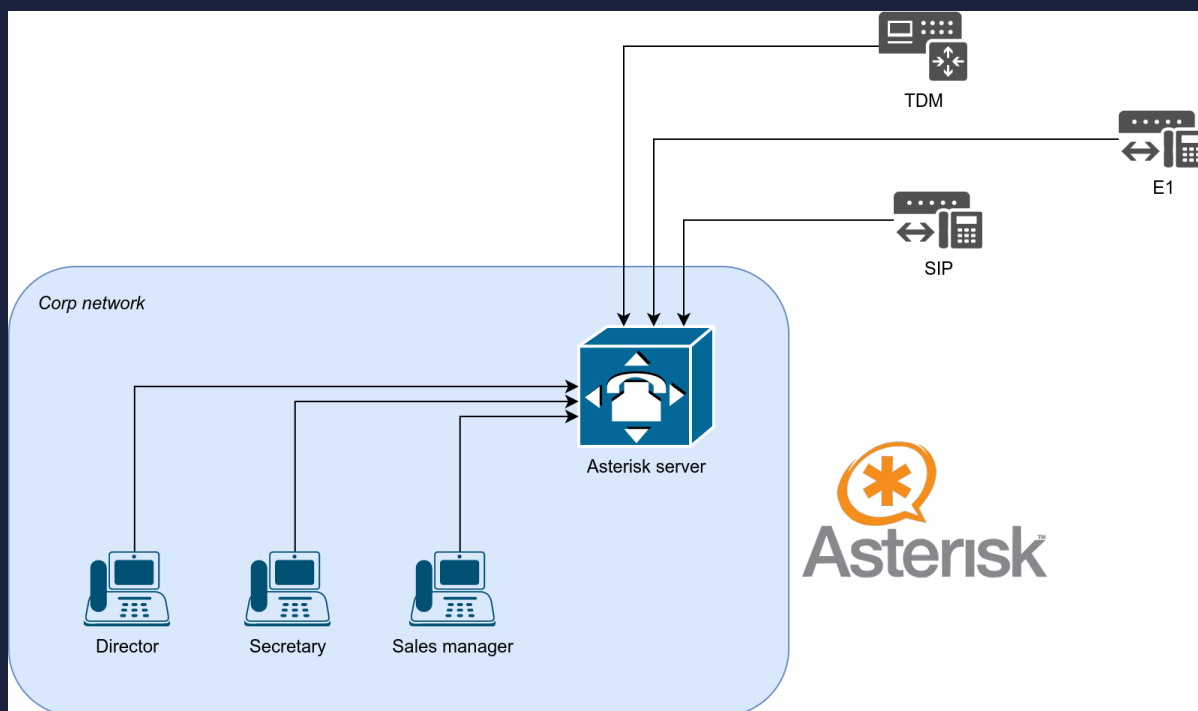
С чего начинались системы телефонии

...некоторые из которых имели платы расширения под цифровые каналы связи



С чего начинались системы телефонии

В итоге многие пришли к тому что проще установить программную АТС (как правило выбор падает на **Asterisk** или его производные)



На этом эволюция телефонии обычно останавливается





102

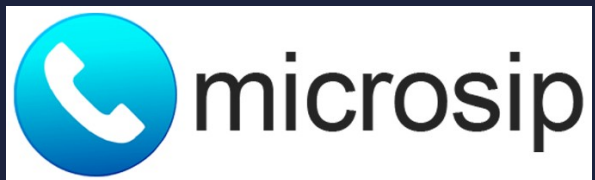
Растет компания —
растут потребности

Бизнес- запросы:



- Не всегда есть возможность поставить аппаратный телефон — нужна «легкая» альтернатива
- Нужно больше контроля (с)
- Все должно быть интегрировано в общую ИТ-инфраструктуру
- Если что-то поменяется — нужно иметь возможность быстро подстроиться
- Другие вопросы из разряда «А что если ...?»

Варианты замены аппаратов:




- Программные телефоны:
 - MicroSIP
 - Linphone
 - Jitsi Desktop
- WebSocket / WebRTC клиенты на основе JS библиотек:
 - SIP5ML
 - JSsip
 - SIPjs

Обеспечение контроля



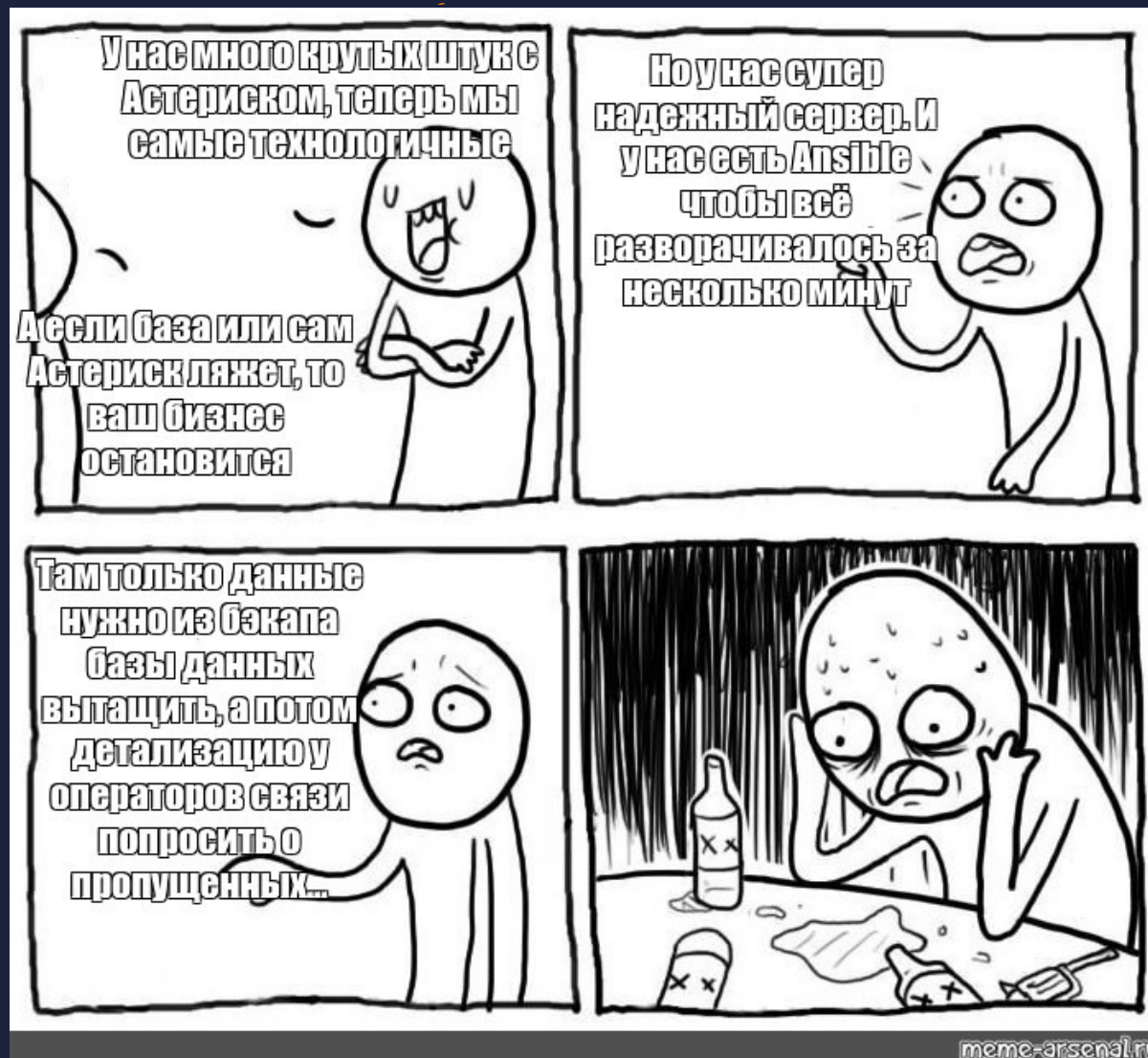
- CDR
- Queue Log
- CEL (иногда он бывает нужен)
- Свои кастомные логи (например на основе событий AMI) — подробнее рассматривалось на AsterConf-2022

Простые интеграции (почти)



- System(), \$SHELL() и т.д.
- Автоматическое конфигурирование телефонов
- Click2Call (например при помощи AMI Originate)


Что там с надежностью?



03

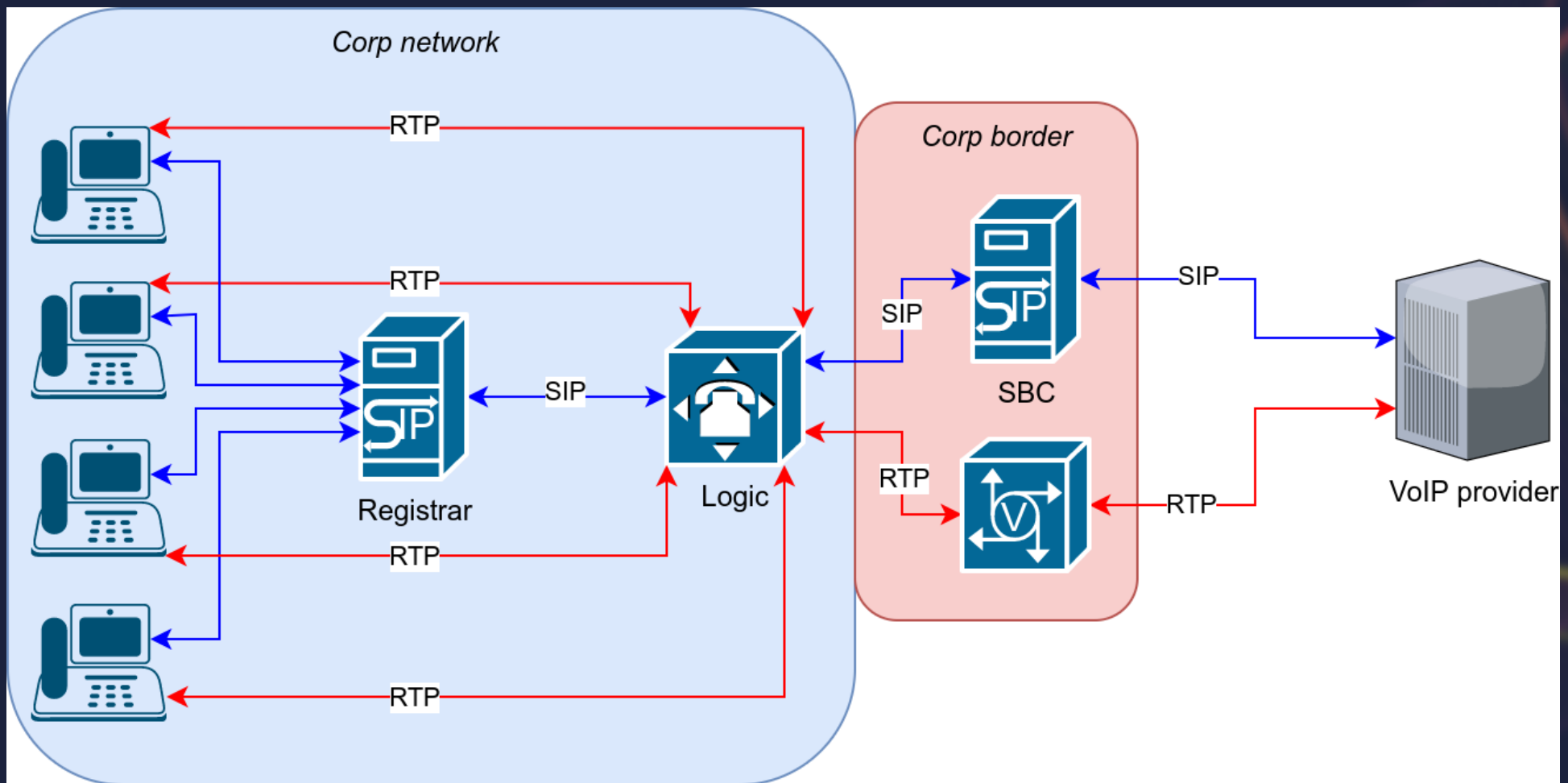
Нюансы архитектуры сервиса

Когда мы доросли до некого «уровня», то:

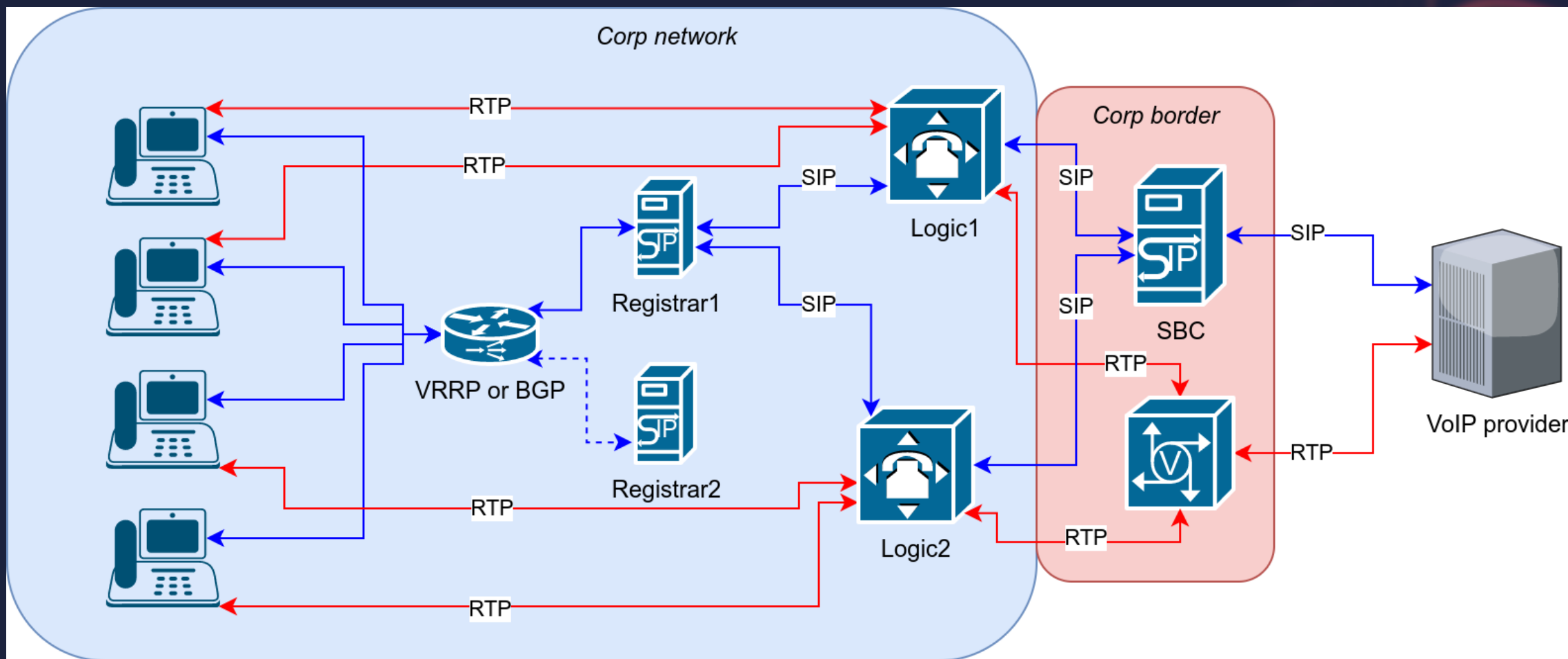


- Сначала вспоминают что база данных важна и её нужно правильно настраивать под конкретные нужды
- Если Asterisk один (пусть даже и с холодным или горячим резервом) — то этого оказывается мало
- Если Asterisk-ов много и они разделены по автономным или частично связанным бизнес-ролям, то это тяжело сопровождать

Дробим на компоненты



Масштабируем компоненты



Что нужно для такого концепта:

- Registrar:
 - VRRP-адрес или BGP-анонсирование (возможно в комбинации с Anycast-схемой)
 - SIP прокси в режиме кластерного регистратора или mid-registrar
 - Балансировка между логическими нодами
- Logic:
 - Если применяются очереди выделить одну или несколько нод под их обработку (в зависимости от бизнес-логики)
 - Если Asterisk-и будут фактическими регистраторами — включить Realtime пиры

Отказоустойчивость SBC

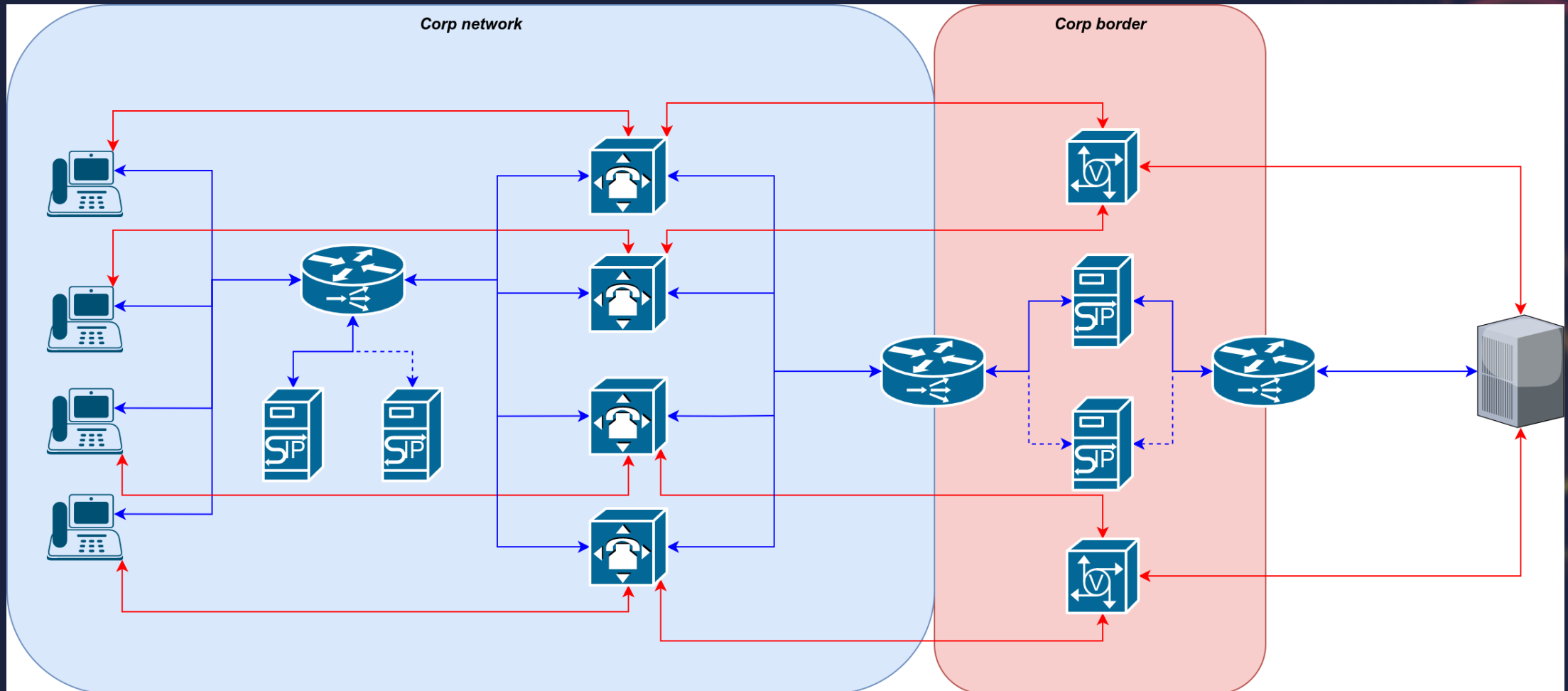


Условно два пути:


- Аналог схемы с Registrar-ом с внутренней и внешней стороны
- Два автономных SBC настроенных на разные учетные записи VoIP провайдеров

Важно: в обоих случаях подразумевается использование кластерного варианта RTPEngine

Итоговая схема претендующая на отказоустойчивость



Что важно сделать (если этого нет):



- SIP мониторинг (Homer 7, SIP3 или аналоги)
- Централизованный сбор логов (предпочтительно в JSON-формате)
- OPTIONS или иные проверки внутренних абонентов и внешних стыков

ИТОГИ



- Универсальной схемы не существует
- Нужно понимать что и где мы хотим улучшить, опираясь на бизнес-требования
- Иногда нужно с чего-то начать и поэтапно модернизировать то что есть (нужна обратная совместимость)

**Спасибо за
внимание!**

Артём Друзь
СберЗдоровье

tg://@Shkiperon

