Chào bạn, tôi là "Call .Net".

Quyết định **"Start small, think big"** (Bắt đầu nhỏ, nghĩ lớn) của bạn là hoàn toàn chính xác. Không ai dại dột đi mua 5-10 con server ngay từ ngày đầu khi chưa viết dòng code nào.

Tôi sẽ thiết kế cho bạn một bộ hồ sơ dự án khởi động (Kick-off Kit) để bạn làm Proof of Concept (PoC) với **1 Server duy nhất**.

### PHẦN 1: HẠ TẦNG & NHÂN SỰ (Giai đoạn PoC)

#### 1. Server cần chuẩn bị (Chỉ cần 1 con)

Để chạy thử FusionPBX (All-in-one: gồm FreeSWITCH, Database, Web Admin), bạn cần thuê 1 VPS/Cloud Server với cấu hình sau:

* **OS:** **Debian 12** (Bắt buộc. FusionPBX chạy mượt nhất trên Debian).
* **CPU:** 4 vCPU (Để compile code FreeSWITCH nhanh hơn).
* **RAM:** 4GB - 8GB.
* **Disk:** 50GB SSD (Ưu tiên NVMe).
* **Network:** Có 1 địa chỉ IP Public tĩnh. Mở các port: 80/443 (Web), 5060/5061 (SIP), 16384-32768 (RTP - Âm thanh).

#### 2. Đội hình Team (Minimal Team)

Với dự án mới, bạn cần tối thiểu 3 vị trí (có thể kiêm nhiệm):

1. **Backend Lead (.NET) - Là bạn:**
   * Thiết kế DB quản lý khách hàng.
   * Viết API Gateway kết nối sang FusionPBX.
   * Nghiên cứu kiến trúc.
2. **VoIP Admin (System Engineer):**
   * Người này chịu trách nhiệm cài đặt FusionPBX lên server Linux.
   * Cấu hình Trunk nhà mạng, test nghe gọi.
   * Debug lỗi SIP khi không có tiếng.
3. **Frontend Dev (React/Vue/Angular):**
   * Làm giao diện cho khách hàng đăng ký, mua số, xem lịch sử cuộc gọi (Vì giao diện gốc của FusionPBX hơi khó dùng cho người dùng cuối).

### PHẦN 2: ROADMAP & TIMELINE (3 THÁNG)

Dưới đây là kế hoạch để biến ý tưởng thành bản MVP (Minimum Viable Product - Sản phẩm khả thi tối thiểu).

#### Tháng 1: Xây dựng nền móng (Infrastructure & Core)

* **Tuần 1:** Thuê Server, Cài đặt FusionPBX (Dùng script install chuẩn). Cấu hình Domain, SSL.
* **Tuần 2:** Đấu nối thử nghiệm Trunk nhà mạng (FPT/CMC/VNPT). Test gọi nội bộ, gọi ra, gọi vào trực tiếp trên FusionPBX (chưa qua .NET).
* **Tuần 3:** Phân tích API của FusionPBX. Dùng Postman chọc thử các API: Tạo Tenant (Domain), Tạo Extension, Lấy CDR (Lịch sử gọi).
* **Tuần 4:** Dựng Project .NET mới. Thiết kế Database nội bộ (Tenant, Users, Pricing).

#### Tháng 2: Phát triển tính năng (Development)

* **Tuần 5:** Code module **Provisioning**: .NET gọi API FusionPBX để tạo tự động Khách hàng & Máy lẻ.
* **Tuần 6:** Code module **Call Control**: Tích hợp ESL (Event Socket) để .NET nhận biết khi nào có cuộc gọi đến/đi.
* **Tuần 7:** Code module **Billing & CDR**: Đồng bộ log cuộc gọi từ FusionPBX về SQL Server của .NET để tính cước.
* **Tuần 8:** Dev Frontend cơ bản (Trang Dashboard, trang cấu hình máy lẻ).

#### Tháng 3: Kiểm thử & Đóng gói (Pilot)

* **Tuần 9:** Test nội bộ (Alpha Test). Cả team dùng hệ thống này để gọi điện hàng ngày.
* **Tuần 10:** Fix bugs (Lỗi mất tiếng, lỗi sai cước, lỗi API timeout).
* **Tuần 11:** Mời 1-2 khách hàng thân thiết dùng thử (Beta Test).
* **Tuần 12:** Đánh giá hiệu năng, lên phương án mở rộng (Add thêm Kamailio nếu cần).

### PHẦN 3: TÀI LIỆU YÊU CẦU PHẦN MỀM (SRS - Simplified)

Tên dự án: NextX Call Center (SaaS Version)

Mục tiêu: Xây dựng hệ thống tổng đài đa khách hàng (Multi-tenant) trên nền tảng FusionPBX.

#### 1. Yêu cầu chức năng (Functional Requirements)

**Module 1: Quản lý Thuê bao (Tenant Management)**

* Hệ thống cho phép tạo mới Tenant (Công ty khách hàng).
* Mỗi Tenant có domain riêng (vd: khachA.call.nextx.vn).
* Mỗi Tenant có giới hạn số lượng Extensions (License).

**Module 2: Quản lý Số & Máy lẻ (Provisioning)**

* Admin có thể gán đầu số (DID) cho Tenant.
* Tenant có thể tự tạo máy lẻ (Extension). Mật khẩu SIP phải sinh tự động và bảo mật.
* Tenant có thể gán đầu số hiển thị (Outbound CID) cho từng máy lẻ.

**Module 3: Định tuyến cuộc gọi (Routing)**

* Cấu hình gọi vào (Inbound): Đổ chuông máy lẻ, nhóm (Ring Group) hoặc lời chào (IVR).
* Cấu hình gọi ra (Outbound): Tự động chọn Trunk giá rẻ nhất.

**Module 4: Báo cáo & Giám sát**

* Xem lịch sử cuộc gọi (CDR) realtime.
* Nghe lại file ghi âm trực tiếp trên Web.
* Giám sát trạng thái máy lẻ (Xanh: Rảnh, Đỏ: Bận, Xám: Offline).

#### 2. Yêu cầu phi chức năng (Non-Functional Requirements)

* **Hiệu năng:** Server 1 Core chịu tải được 20 cuộc gọi đồng thời (để test).
* **Độ trễ (Latency):** < 150ms (để đảm bảo chất lượng thoại).
* **Bảo mật:** API giao tiếp giữa .NET và FusionPBX phải có Token. Chặn IP lạ đăng nhập Admin.

### PHẦN 4: TÀI LIỆU THIẾT KẾ HỆ THỐNG (SDS - Simplified)

#### 1. Kiến trúc tổng thể (Architecture Diagram)

Đoạn mã

Client Apps (Web/Mobile)  
 │  
 ▼  
[ .NET 8 API Gateway ] <───> [ SQL Server / Redis ]  
 │ (REST API / ESL)  
 │  
 ▼  
[ FusionPBX Server (All-in-One) ]  
 ├── Nginx (Web Server)  
 ├── PHP-FPM (FusionPBX Core)  
 ├── FreeSWITCH (SIP Engine)  
 └── PostgreSQL (PBX DB)

#### 2. Thiết kế Database (High-level Schema)

Bạn cần map dữ liệu giữa .NET và FusionPBX:

**Bảng Tenants (Trong .NET)**

* Id (PK)
* Name: Tên công ty
* FusionDomainUuid: UUID của domain bên FusionPBX (Khóa ngoại logic)
* Status: Active/Inactive

**Bảng Extensions (Trong .NET)**

* Id (PK)
* TenantId (FK)
* ExtensionNumber: 101, 102
* SipPassword: Lưu trữ để hiển thị lại cho user
* FusionExtensionUuid: UUID của extension bên FusionPBX

#### 3. Công nghệ sử dụng (Tech Stack)

* **Backend:** .NET 8.0 (Clean Architecture).
* **Database:**
  + Nghiệp vụ: SQL Server (hoặc PostgreSQL cho đồng bộ).
  + Cache: Redis (để lưu trạng thái cuộc gọi realtime).
* **Communication:**
  + API: HttpClient (gọi FusionPBX REST API).
  + Realtime: NEventSocket (kết nối FreeSWITCH Event Socket).
* **VoIP Core:** FusionPBX v5.x + FreeSWITCH v1.10.x.
* **OS:** Debian 12 Bookworm.

### Lời khuyên bắt đầu ngay hôm nay

Bạn không cần viết hết đống tài liệu này ra Word đâu.

1. **Việc 1:** Thuê ngay 1 con VPS Debian 12 (khoảng 200k-300k/tháng).
2. **Việc 2:** Tìm từ khóa *"Install FusionPBX Debian 12 script"* và chạy lệnh cài đặt.
3. **Việc 3:** Khi cài xong, đăng nhập vào web admin của nó, bạn sẽ thấy "chân trời mới" khác hẳn FreePBX.

Đây là bước đi an toàn, tốn ít chi phí nhất nhưng lại học được nhiều nhất. Chúc bạn khởi động dự án thành công!