

```

.
├── api                               # 1. PROTOCOL DEFINITIONS (Contracts)
│   └── proto
│       └── v1
│           ├── admin.proto
│           ├── auth.proto
│           ├── chats.proto
│           ├── cron.proto
│           ├── files.proto
│           ├── notification.proto
│           └── subscription.proto
│               # Versioning for future API changes
│               # Admin specific gRPC methods
│               # User/Login/Registration
│               # Real-time message APIs
│               # Internal scheduler/job APIs
│               # File management/upload/download
│               # Push/Email/SMS events
│               # Payments/Plans
└── cmd                                # 2. ENTRY POINTS (Runnable Binaries)
    ├── api-gateway
    │   └── main.go
    ├── auth-service
    │   └── main.go
    ├── chat-service
    │   └── main.go
    ├── cron-service
    │   └── main.go
    └── file-service
        └── main.go
            # HTTP/REST entry point (translates to gRPC)
            # Auth gRPC server
            # Chat gRPC server
            # Cron gRPC server (for background tasks)
            # Files gRPC server
└── configs                            # 3. SERVICE-SPECIFIC CONFIGURATION
    ├── auth.yaml
    ├── chat.yaml
    ├── cron.yaml
    ├── files.yaml
    └── gateway.yaml
        # DB credentials, JWT secret, OAuth keys
        # Redis connection, channel limits
        # S3 credentials, file limits
        # Ports, target gRPC service addresses
└── docker
    └── docker-compose.yml
        # Defines Postgres, Redis, MinIO, etc.
└── internal                           # 4. PRIVATE BUSINESS LOGIC (Decoupled Core)
    # --- Auth Service Module ---
    └── auth
        ├── domain
        │   └── user.go
        ├── port
        │   ├── repository.go
        │   └── service.go
        ├── app
        │   ├── service.go
        │   └── oauth_logic.go
        └── adapter
            ├── storage
            │   └── postgres
            │       └── user_repo.go
            └── grpc
                └── server.go
            # Pure Go structs (User, Session). No external dep
            # Interfaces defining contracts. Used for DI/Mocki
            # interface UserRepository {...}
            # interface AuthService {...}
            # Business Logic (Use Cases). Implements service.go
            # Register/Login/Validate logic
            # Specific Google/3rd party integration
            # Infrastructure/External Implementations
    # --- Files Service Module ---
    └── files
        ├── domain
        │   # File, Bucket, AccessControl structs
        ├── port
        │   └── file_repository.go
        ├── app
        │   └── file_service.go
        │       # Upload/Download logic
        └── adapter
            ├── storage
            │   └── s3_storage.go
            # Handles connection to S3/MinIO

```

```

    |   └── grpc
    |       └── server.go
# --- API Gateway Module ---
└── gateway
    ├── handlers
    │   ├── auth_handler.go
    │   └── file_handler.go
    ├── middleware
    │   └── auth_middleware.go
    └── routes.go

# The REST/HTTP Layer
# HTTP endpoints, calls internal gRPC clients
# Handles /v1/register, /v1/login
# Handles /v1/files/upload
# HTTP middleware (CORS, RequestID, JWT validation)
# Defines the routing table

migrations
├── auth
│   ├── 001_create_users.up.sql
│   └── 001_create_users.down.sql
└── files
    ├── 001_file_metadata.up.sql
    └── 001_file_metadata.down.sql

# 5. DATABASE MIGRATIONS (Separated by DB/Service)
# Schema for the auth-service DB
# Schema for the file-service DB

pkg
├── constant
├── db
│   └── postgres.go
├── errors
│   ├── app_error.go
│   └── grpc_error.go
├── logger
│   └── logger.go
├── pb
│   ├── auth
│   │   └── auth_grpc.pb.go
│   └── files
│       └── files_grpc.pb.go
└── utils
    ├── password.go
    └── validation.go

# 6. PUBLIC UTILITIES (Shared Libraries)
# Shared application constants
# Generic DB connection factory (Postgres)

# Generic error wrappers/codes

# Wrapper around Zap/Logrus

# GENERATED GO CODE (The only place contracts are
# defined)

# Generic non-business tools (hashing, validation)

.gitignore
go.mod
go.sum
Makefile
# Automation for proto gen, migrations, build
README.md

```