

# Charting Website Tiered Architecture

## \*\*Tier Overview\*\*

...

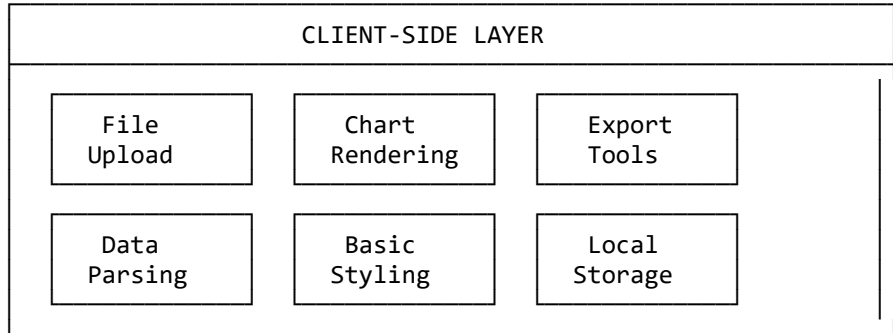
NOOB (Anonymous)	REGISTERED (Free)	VIPER (\$3/month)
<ul style="list-style-type: none"><li>• Basic Plotly</li><li>• Client-side</li><li>• Watermarked</li><li>• IP-limited</li></ul>	<ul style="list-style-type: none"><li>• Enhanced</li><li>• Hybrid</li><li>• Watermarked</li><li>• Session-limit</li></ul>	<ul style="list-style-type: none"><li>• Premium</li><li>• Server-heavy</li><li>• Branded</li><li>• Unlimited</li></ul>

...

## \*\*Architecture Layers

### \*\*1. Client-Side Layer (All Tiers)

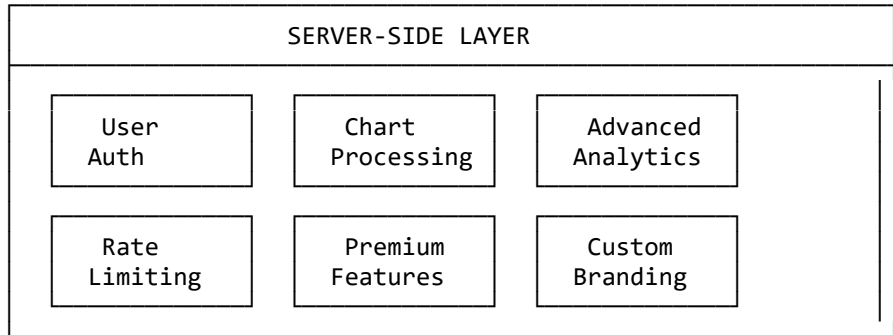
...



...

### \*\*2. Server-Side Layer (Tier-Dependent)

...

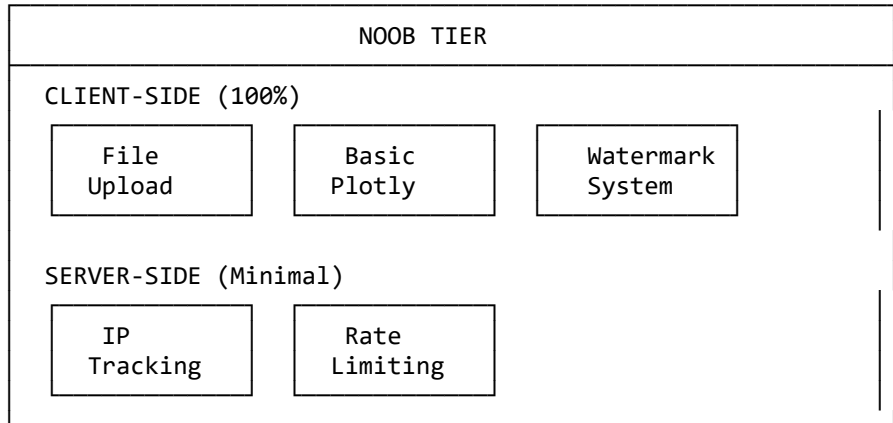


...

## \*\*Tier-Specific Architecture

### \*\*NOOB Tier Architecture

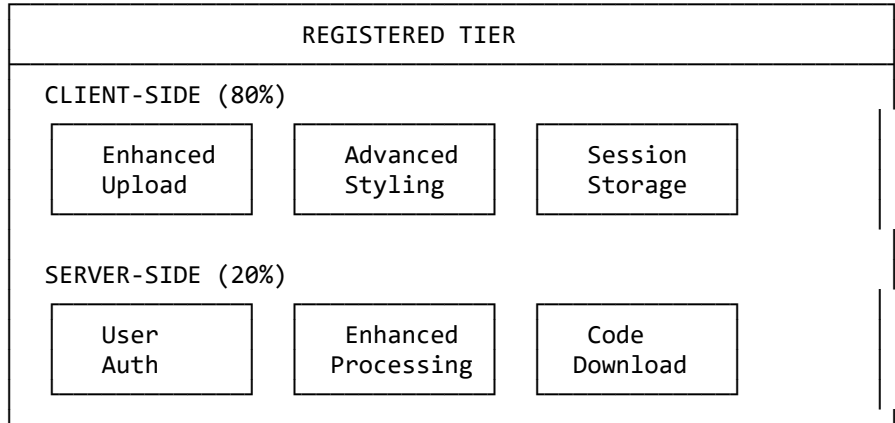
...



...

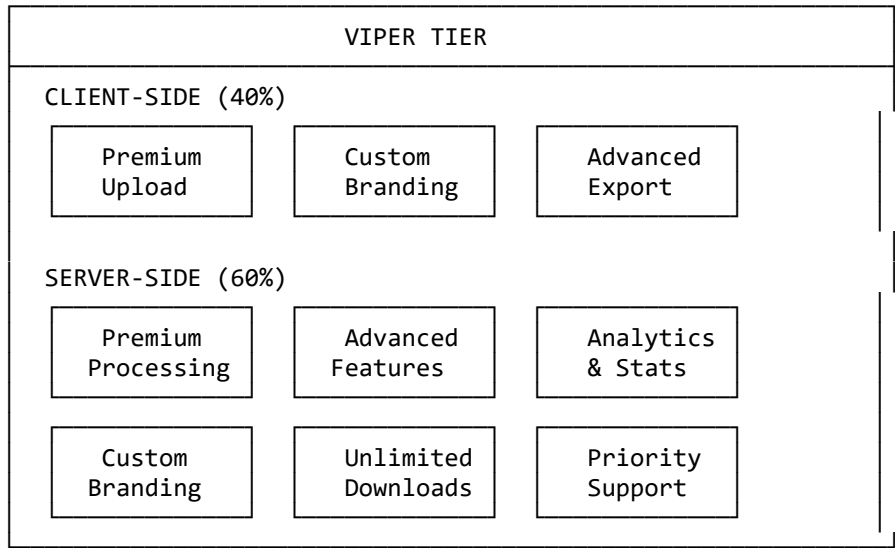
- \*\*Server Calls:\*\***
- `POST /api/track-usage` - IP-based usage tracking
  - `GET /api/limits` - Check monthly limits
  - `POST /api/watermark` - Generate watermarks

**### \*\*REGISTERED Tier Architecture\*\***



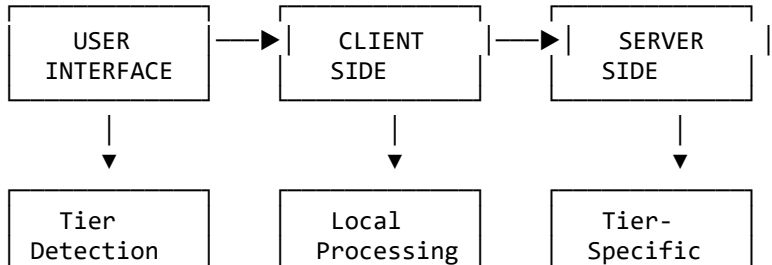
- \*\*Server Calls:\*\***
- `POST /api/auth/login` - User authentication
  - `POST /api/process-enhanced` - Enhanced chart processing
  - `GET /api/download-code` - Code generation (limited per session)
  - `POST /api/save-preferences` - User preferences

**### \*\*VIPER Tier Architecture\*\***



- \*\*Server Calls:\*\***
- `POST /api/premium/process` - Premium chart processing
  - `POST /api/branding/apply` - Custom branding application
  - `GET /api/premium/features` - Advanced chart types
  - `POST /api/analytics/track` - Usage analytics
  - `GET /api/premium/downloads` - Unlimited downloads

**## \*\*Data Flow Architecture\*\***





```
### **Performance Strategy**
- **Caching**: Server-side caching for common operations
- **CDN**: Static assets served via CDN
- **Load Balancing**: Distribute server load across instances
- **Database**: Efficient queries, connection pooling
```

## \*\*Implementation Roadmap\*\*

```
### **Phase 1: NOOB Tier**
- Implement IP-based tracking
- Basic client-side functionality
- Watermark system
- Rate limiting
```

```
### **Phase 2: REGISTERED Tier**
- User authentication system
- Enhanced features
- Session-based limits
- Code download system
```

```
### **Phase 3: VIPER Tier**
- Premium features
- Custom branding
- Analytics system
- Unlimited downloads
```

## \*\*Technical Requirements\*\*

```
### **Frontend Technologies**
- HTML5, CSS3, JavaScript (ES6+)
- Plotly.js for charting
- Local storage for session management
- Canvas API for watermarking
```

```
### **Backend Technologies**
- Python Flask/Django or Node.js
- PostgreSQL/MongoDB for data storage
- Redis for caching and session management
- AWS/GCP for cloud hosting
```

```
### **Security Measures**
- JWT tokens for authentication
- Rate limiting middleware
- Input validation and sanitization
- HTTPS encryption
- CORS configuration
```

This architecture provides a robust, scalable foundation that protects code while delivering appropriate functionality for each tier level.