

# Architectural Drivers Document ( Demo )

## ระบบที่เลือก (Target System)

ชื่อระบบที่กลุ่มเลือก: Google Translate ประเภท SaaS (Software as a Service)

ประเภทระบบ:  Chat / Messaging  E-Commerce  Streaming (Video/Music)   
Food Delivery  อื่น ๆ ระบุ: SaaS (Software as a Service)

### System Description

ระบบ Google Translate เป็นบริการแปลภาษาอัตโนมัติที่ให้บริการแบบ Software as a Service (SaaS) โดยมีวัตถุประสงค์เพื่อลดอุปสรรคทางภาษาสำหรับผู้ใช้งานทั่วโลก ระบบนี้เป็นที่น่าสนใจในการศึกษาเนื่องจากต้องมีการจัดการข้อมูลหลายรูปแบบ เช่น ข้อความ (Text), เสียง (Voice), และรูปภาพ (Image)

หน้าที่หลักของระบบคือการรับ Input จากผู้ใช้ (ไม่ว่าจะเป็นการพิมพ์ข้อความ, การพูดผ่านไมโครโฟน, หรือการอัปโหลดรูปภาพ) เพื่อทำการแปล และส่งมอบ Output ผลลัพธ์การแปลที่ถูกต้องกลับไปยังผู้ใช้ ตัวระบบเองทำหน้าที่เป็นศูนย์กลางจัดการ Workflow (Backend) โดยพึ่งพาการเรียกใช้ External Services (เช่น OCR และ Speech Engine) เพื่อประมวลผลข้อมูลประเภทภาษา/เสียง นอกจากนี้ยังต้องจัดการระบบจัดเก็บประวัติการแปล (History) และคำศัพท์/วลีโปรด (Phrasebook) ที่ต้องซิงค์ข้อมูลข้ามอุปกรณ์ผ่านระบบ Google Account

## Target Users

- General User (ผู้ใช้งานทั่วไป): ผู้ที่ต้องการแปลภาษาเพื่อทำความเข้าใจหรือสื่อสารในชีวิตประจำวัน โดยใช้ Input หลากหลายรูปแบบ (ข้อความ, เสียง, รูปภาพ)
- Contributor (ผู้ร่วมแก้ไขข้อมูล): ผู้ใช้ที่มีความเชี่ยวชาญทางภาษา ทำหน้าที่ตรวจสอบและเสนอแนะคำแปลที่ถูกต้องเพื่อปรับปรุงคุณภาพของระบบแปล
- System Admin (ผู้ดูแลระบบ): ผู้รับผิดชอบในการจัดการบัญชีผู้ใช้, ดูแล Server, จัดการ Dashboard สถิติการใช้งาน, และเพิ่ม/ลบภาษาที่ระบบรองรับ
- 

## Key Features (Top 10)

- Text Translation: แปลข้อความที่ผู้ใช้พิมพ์หรือวาง (Paste) ทันที
- Voice Translation: แปลคำพูดที่ผู้ใช้พูดใส่ไมโครโฟน โดยใช้ Speech-to-Text Engine
- Image Translation (OCR): แปลข้อความจากรูปภาพที่อัปโหลดหรือถ่ายด้วยกล้อง โดยใช้ OCR/Google Lens
- Text-to-Speech (TTS): อ่านออกเสียงคำแปลที่ได้ให้ผู้ใช้งาน
- History Management: จัดเก็บประวัติการแปลที่ผ่านมาของผู้ใช้งาน
- Phrasebook Management (Saved/Favorites): อนุญาตให้ผู้ใช้บันทึกคำแปลหรือวลีที่ใช้บ่อย
- Data Synchronization: ซิงค์ข้อมูล History และ Phrasebook ข้ามอุปกรณ์ ( เช่น จาก PC ไป Mobile ) ผ่าน Google Account
- Quality Feedback (Suggest Edit): อนุญาตให้ Contributor เสนอแนะหรือยืนยันคำแปลที่ถูกต้อง
- Language Management: System Admin สามารถเพิ่มหรือลบภาษาที่ระบบรองรับได้
- System Monitoring: System Admin สามารถดู Dashboard สถิติการใช้งานของระบบได้

# Functional Requirements

## User Management

- FR-01: ผู้ใช้สามารถ Register และ Login ผ่าน Google Account System ได้
- FR-02: ระบบต้องสามารถดึงข้อมูล Phrasebook และ History ที่เคยบันทึกไว้ของผู้ใช้มาแสดงผลได้ (Sync)
- FR-03: System Admin สามารถดู Dashboard สถิติการใช้งาน และจัดการบัญชีผู้ใช้ได้
- FR-04: ระบบต้องกำหนดสิทธิ์การเข้าถึงตามบทบาท (Role-Based Access) ระหว่าง User, Admin, และ Contributor ได้อย่างชัดเจน

## Core Features

- FR-05: ระบบสามารถรับ ข้อความ (Text) เพื่อส่งไปประมวลผลการแปล และแสดงผลลัพธ์ที่แปลแล้ว
- FR-06: ระบบสามารถรับ ไฟล์เสียง (จากการพูดของผู้ใช้) เพื่อส่งไปให้ Speech-to-Text Engine ประมวลผลก่อนส่งไปแปล
- FR-07: ระบบสามารถรับ รูปภาพ (จากการอัปโหลด/กล้อง) เพื่อส่งไปให้ OCR/Vision API ดึงข้อความก่อนส่งไปแปล
- FR-08: ระบบสามารถส่งข้อความที่แปลเสร็จแล้วไปให้ Text-to-Speech Engine เพื่อแปลงเป็นเสียงให้ผู้ใช้ฟังได้
- FR-09: ผู้ใช้สามารถเพิ่มคำแปล/วลีที่ชอบลงใน Phrasebook ได้
- FR-10: ผู้ใช้สามารถดู ประวัติการแปล (History) ของตัวเองได้

## Data Quality & Feedback

- FR-11: Contributor สามารถกด "Suggest Edit" เพื่อเสนอแนะคำแปลที่ถูกต้องได้
- FR-12: Contributor สามารถช่วยยืนยันความหมาย ของคำศัพท์ใหม่ ๆ ได้
- FR-13: System Admin สามารถเพิ่ม/ลบ ภาษาที่รองรับ ในระบบได้

## System Interoperability

- FR-14: ระบบต้องสามารถส่งภาพไปยัง OCR/Vision API และรับ "ข้อความที่แกะได้" กลับมาได้
- FR-15: ระบบต้องสามารถส่งไฟล์เสียงไปยัง Speech-to-Text Engine และรับ "ข้อความ (String)" กลับมาได้
- FR-16: ระบบต้องสามารถส่งข้อความไปยัง Text-to-Speech Engine และรับ เสียงที่สังเคราะห์ (Synthesized Voice) กลับมาได้

## Quality Attributes

### QA-1: Performance (Latency)

Scenario: การแปลงข้อความที่ผู้ใช้พิมพ์

ส่วน	รายละเอียด
Source	General User (ผู้ใช้งานทั่วไป)
Stimulus	ผู้ใช้พิมพ์ข้อความความยาว 50 ตัวอักษร (Text Input) ในช่องแปลงภาษา

Artifact	Core Translation Engine (API Gateway และ Internal Translation Service)
Environment	ภายใต้สภาพการทำงานปกติ (Average Load: 5,000 requests/second)
Response	ระบบทำการแปลงและแสดงผลลัพธ์การแปลงบน User Interface
Response Measure	เวลาตั้งแต่การกดปุ่มสุดท้ายจนถึงการแสดงผลลัพธ์บนหน้าจอ (End-to-End Latency) ต้อง น้อยกว่า 500 มิลลิวินาที (ms) สำหรับ 95% ของ Requests

**คำอธิบายแบบเต็ม:** เนื่องจาก Google Translate เป็นระบบที่เน้นการโต้ตอบแบบเรียลไทม์ (Real-time) การตอบสนองที่รวดเร็วจึงมีความสำคัญสูงสุด โดยเฉพาะสำหรับการแปลงข้อความที่พิมพ์ ผู้ใช้คาดหวังผลลัพธ์ที่ปราฏภันท์เมื่อพิมพ์เสร็จ ดังนั้น ระบบจึงต้องได้รับการออกแบบให้มี **High Concurrency** และมี Latency ต่ำ โดยเฉพาะในการเข้ามต่องกับ Translation Engine ภายใน

## QA-2: Scalability (Horizontal Scaling)

Scenario: การรองรับผู้ใช้งานพร้อมกัน

ส่วน	รายละเอียด
Source	General User ทั่วโลก
Stimulus	จำนวนผู้ใช้งานพร้อมกันเพิ่มขึ้นจาก 10,000 เป็น 100,000 Requests ต่อวินาที ในช่วงเหตุการณ์สำคัญระดับโลก
Artifact	Core API Gateway และ History Management Service
Environment	ในสภาพ Peak Load ที่สูงขึ้น 10 เท่า
Response	ระบบต้องสามารถประมวลผลการแปลง และจัดการการบันทึก History ของผู้ใช้ใหม่ทั้งหมดได้
Response Measure	Latency ในการประมวลผลคำขอ (เช่น การแปลงข้อความ, การบันทึก History) ต้องไม่เพิ่มขึ้นเกิน 25% และ Throughput ต้องเพิ่มขึ้น 900%

คำอธิบายแบบเต็ม: Google Translate เป็นระบบที่มีผู้ใช้หลายล้านคนทั่วโลก และมีโอกาสเกิด Traffic สูงมากในช่วงเวลาเดียวกัน (High Concurrency) สถาปัตยกรรมจึงต้องเป็นแบบ **Distributed System** ที่สามารถเพิ่มทรัพยากร (เช่น Server) แบบ Horizontal Scaling ได้ง่าย

และรวดเร็ว โดยเฉพาะส่วนของ API Gateway และ History Management เพื่อให้สามารถรองรับคำขอจำนวนมหาศาลได้โดยที่ประสิทธิภาพ (Latency) ไม่ลดลงอย่างมีนัยสำคัญ

### QA-3: Availability (Fault Tolerance)

Scenario: External Service ขัดข้อง

ส่วน	รายละเอียด
Source	External Service เช่น Speech-to-Text (STT) Engine
Stimulus	STT Engine ไม่ตอบสนอง เนื่องจากเกิดความผิดพลาดในการเชื่อมต่อเครือข่าย
Artifact	Google Translate System (API Gateway)
Environment	สภาพการใช้งานปกติ
Response	ระบบต้องแจ้งเตือนผู้ใช้ว่า "Voice Input ไม่พร้อมใช้งานชั่วคราว" และยังคงอนุญาตให้ใช้งานฟังก์ชันการแปล Text Input และ History/Phrasebook ได้ตามปกติ

Response Measure	Text Translation Feature ต้องยังคงมี Uptime > 99.99% และ มี Latency เพิ่มขึ้น ไม่เกิน 10%
------------------	---

คำอธิบายแบบเต็ม: ระบบ Google Translate พึ่งพาบริการภายนอก (External Services) หลายตัว (OCR, Speech, Identity) ในการประมวลผลฟังก์ชันหลัก หากบริการใดบริการหนึ่งล้มเหลว ระบบต้องถูกออกแบบมาให้มีการแยกส่วน (Decoupling) เพื่อให้ฟังก์ชันอื่น ๆ ที่ไม่เกี่ยวข้องกับบริการที่ล้มเหลวยังคงใช้งานได้ (Graceful Degradation) เพื่อรักษาความต่อเนื่องของบริการหลัก (Text Translation)

#### QA-4: Maintainability (Modifiability)

Scenario: การเพิ่มภาษาใหม่

ส่วน	รายละเอียด
Source	System Admin หรือ Developer Team
Stimulus	มีการขอให้ เพิ่มภาษาใหม่ (เช่น ภาษาท้องถิ่นที่เพิ่งได้รับการสนับสนุน) เข้าสู่ระบบแปล
Artifact	Language Dataset และ Core System Logic
Environment	Development/Staging Environment

Response	Developer สามารถเพิ่ม Dataset ภาษาใหม่และปรับปรุง Configuration ได้ โดยไม่กระทบต่อภาษาเดิมที่รองรับ
Response Measure	ระยะเวลาในการ Deploy ภาษาใหม่ ตั้งแต่เริ่มต้นจนถึงใช้งานจริง (Go-Live) ต้อง น้อยกว่า 48 ชั่วโมง

**คำอธิบายแบบเต็ม:** เนื่องจากระบบต้องมีการดูแล Server และจัดการ Dataset ภาษาต่าง ๆ สถาปัตยกรรมควรเอื้อต่อการปรับปรุง (Modifiability) โดยเฉพาะอย่างยิ่งในการเพิ่มภาษาหรือ การอัปเดตโมเดลการแปล โดยที่การเปลี่ยนแปลงในส่วนหนึ่งไม่ส่งผลกระทบต่อส่วนอื่น ๆ

#### QA-5: Security (Authentication/Authorization)

**Scenario:** การเข้าถึงข้อมูล History

ส่วน	รายละเอียด
Source	General User (User A)
Stimulus	User A พยายามใช้ Auth Token ของ User B เพื่อ เข้าถึง History ของ User B

Artifact	Identity Service และ History Management Service
Environment	สภาพการใช้งานปกติ
Response	ระบบต้องปฏิเสธการเข้าถึงข้อมูล History ของ User B โดย User A และบันทึกเหตุการณ์ความพยายามในการเข้าถึงที่ผิดปกติ
Response Measure	อัตราการป้องกันการเข้าถึง ข้อมูลส่วนตัวโดยไม่ได้รับอนุญาตต้องเป็น 100% (Zero Tolerance) และมีการแจ้งเตือนความพยายามในการลงทะเบียน

**คำอธิบายแบบเต็ม:** ระบบต้องมีการจัดการบัญชีผู้ใช้กลางของ Google และจัดการ Logic/History ข้อมูล History และ Phrasebook ถือเป็นข้อมูลส่วนตัวของผู้ใช้ ดังนั้นการควบคุมการเข้าถึง (Access Control) โดยใช้ Auth Token และการกำหนดสิทธิ์ (Authorization) ตามบทบาท (Role-Based Access) ต้องเข้มงวดเป็นพิเศษ เพื่อรักษาความลับและความสมบูรณ์ของข้อมูล

#### QA-6: Usability (Learnability/Intuitiveness)

**Scenario:** การใช้ฟังก์ชันแปลภาษา (OCR)

ส่วน	รายละเอียด
Source	General User ที่ใช้งานครั้งแรก

Stimulus	ผู้ใช้กดปุ่ม "กล้อง (Camera)" เพื่อแปลงความจากรูปภาพที่อยู่ตรงหน้า <sup>49</sup>
Artifact	User Interface (Web/App) <sup>5050</sup>
Environment	สภาพแวดล้อมจริง (Mobile App)
Response	ระบบต้องแสดงคำแนะนำและขั้นตอนการใช้กล้องอย่างชัดเจน และ ประมวลผลภาพอย่างรวดเร็วเพื่อให้ผู้ใช้ได้รับผลลัพธ์
Response Measure	ผู้ใช้ที่ใช้งานครั้งแรกสามารถใช้งานฟังก์ชันแปลงภาพได้ สำเร็จภายใน 3 คลิก/แตะ โดยใช้เวลาเฉลี่ย ไม่เกิน 15 วินาที

**คำอธิบายแบบเต็ม:** Google Translate เน้นการใช้งานที่หลากหลายรูปแบบ (Text, Voice, Image) และต้องอำนวยความสะดวกในการใช้งานให้แก่ผู้ใช้ทั่วไป ดังนั้น User Interface (Web/App) ต้องมีความง่ายและเป็นธรรมชาติ (Intuitiveness) เพื่อให้ผู้ใช้สามารถเข้าถึงฟังก์ชันที่ซับซ้อน เช่น การแปลงภาพ (OCR) ได้โดยง่ายและรวดเร็ว

## Constraints

### Technical Constraints

- **Constraint 1:** ต้องใช้บริการ Google Account System (Identity Service) สำหรับการจัดการบัญชีผู้ใช้และการซิงค์ข้อมูล (History/Phrasebook)
- **Constraint 2:** ต้องพึ่งพา External Services/APIs ของ Google เอง (เช่น OCR/Vision API, Speech-to-Text Engine, Text-to-Speech Engine) ในการประมวลผลข้อมูลภาพและเสียง
- **Constraint 3:** ต้องรองรับการทำงานบน หลายแพลตฟอร์ม (Web/App) และการซิงค์ข้อมูลข้ามอุปกรณ์

### Time Constraints

- **Constraint 1:** ต้องจัดเก็บ History และ Phrasebook ของผู้ใช้ในรูปแบบที่สามารถค้นหาและดึงข้อมูลมาแสดงผลได้อย่างรวดเร็วมาก เพื่อรับ Latency ตามกำหนดของ Performance

### Budget Constraints

- **Constraint 1:** งบประมาณอาจถูกจำกัดด้วย ค่าใช้จ่ายในการเรียกใช้ External APIs (OCR, Speech) เนื่องจากเป็นฟังก์ชันหลักที่ใช้บ่อย

### Legal/Policy Constraints

- **Constraint 1:** ต้องปฏิบัติตามนโยบายความเป็นส่วนตัว (Privacy Policy) ของ Google ในการจัดเก็บและใช้ข้อมูลส่วนตัวของผู้ใช้ (เช่น History)

## Assumptions

1. **Assumption 1:** External Services (OCR, Speech Engine) ที่เรียกใช้ มีความน่าเชื่อถือ (Reliability) และมี Uptime สูงเพียงพอ
2. **Assumption 2:** Core Translation Engine (ซึ่งเป็นบริการภายในที่ระบบเรียกใช้) มีคุณภาพการแปลที่แม่นยำและรวดเร็วตามที่ต้องการ
3. **Assumption 3:** การจัดการ History ของผู้ใช้งานนั้นไปที่การจัดเก็บและดึงข้อมูลที่เกี่ยวข้องกับข้อความ (Text-based History)
4. **Assumption 4:** ระบบสามารถแยกส่วน (Decouple) ระหว่าง Client Interface (Web/App) และ Core API Gateway/Backend ได้อย่างสมบูรณ์
5. **Assumption 5:** Contributor เป็นผู้ใช้งานที่มีความเชี่ยวชาญทางภาษาจริง และให้ข้อมูล Feedback ที่ถูกต้องและเป็นประโยชน์

## Quality Attributes Priority

Rank	Quality Attribute	เหตุผล
1	Performance	ระบบแปลภาษาต้องตอบสนองแบบเรียลไทม์ ผู้ใช้คาดหวังผลลัพธ์ทันที Latency ต่ำเป็นหัวใจหลักของประสบการณ์ผู้ใช้
2	Scalability	ต้องรองรับผู้ใช้งานพร้อมกันจำนวนมาก (High Concurrency) และรองรับการเพิ่มขึ้นของ Traffic ทั่วโลก

3	Availability	เนื่องจากพื้นที่บริการภายนอกหลายตัว ระบบต้องทนทานต่อความล้มเหลว (Fault Tolerance) และรักษาความต่อเนื่องของบริการหลัก (Text Translation)
4	Security	ต้องมีการควบคุมการเข้าถึงข้อมูลส่วนตัว (History/Phrasebook) อย่างเข้มงวดผ่าน Google Account System
5	Usability	ระบบต้องใช้งานง่ายสำหรับพิมพ์ชัน Input ที่หลากหลาย (Text, Voice, Image) เพื่อให้ผู้ใช้ทั่วไปสามารถเข้าถึงได้
6	Maintainability	การเพิ่มภาษาใหม่หรือปรับปรุงระบบควรทำได้จ่าย แต่ความสำคัญจะน้อยกว่า 5 ข้อแรกสำหรับระบบที่ตั้งตัวมานาน

## Trade-offs Analysis

### Trade-off #1: Performance vs. Data Consistency (History Sync)

อธิบาย: เพื่อเพิ่ม Performance และ Scalability ในการจัดการ History และ Phrasebook ของผู้ใช้งานหลายล้านคน การเขียนข้อมูล (Write) อาจไม่ได้เกิดขึ้นพร้อมกันบนทุก Server ในทันที (Eventual Consistency)

#### Decision:

เลือก Eventual Consistency สำหรับ History/Phrasebook เนื่องจาก Performance และ Scalability มีความสำคัญมากกว่า ในการรองรับ High Concurrency ผู้ใช้ยอมรับได้หาก History บนอุปกรณ์ใหม่ใช้เวลาซิงค์ 2-3 วินาที (Latency ยอมรับได้เพิ่มขึ้นเล็กน้อยเพื่อแลกกับ Throughput ที่สูงขึ้น) แต่ Strong Consistency ยังคงจำเป็นสำหรับข้อมูลที่สำคัญมาก เช่น ข้อมูลบัญชีผู้ใช้ (Identity Service)

### Trade-off #2: Availability vs. Integration Complexity

อธิบาย: การรักษา Availability ให้สูง (QA-3) จำเป็นต้องมีกลไกป้องกันความล้มเหลวจาก External Services (OCR, Speech Engine) ซึ่งหมายความว่า Core API Gateway ต้องมีการเพิ่ม Logic ที่ซับซ้อน เช่น Circuit Breaker, Retry Mechanisms, และ Fallback Logic ซึ่งทำให้ Maintainability (Modifiability) และความซับซ้อนในการรวมระบบ (Integration Complexity) เพิ่มขึ้น

#### Decision:

เลือกเพิ่ม Integration Complexity (ลด Maintainability) เพื่อรักษา Availability ให้อยู่ในระดับสูง (เช่น 99.99%) เพราะความล้มเหลวของ External Service ส่งผลกระทบโดยตรงต่อฟังก์ชันหลักของระบบ (เช่น การแปลงจากเสียง/ภาพ) ดังนั้นการเพิ่ม Logic เพื่อป้องกันความล้มเหลวจึงเป็นสิ่งที่จำเป็น

---

## Trade-off #3: Security (Strict Access Control) vs. Usability (Contributor Workflow)

อธิบาย: เพื่อให้มั่นใจใน Security และการกำหนดสิทธิ์ตามบทบาท (Role-Based Access) ระบบอาจต้องใช้ขั้นตอนการยืนยันตัวตนและการอนุญาตที่เข้มงวดมากขึ้นสำหรับ Contributor ที่ต้องการ "Suggest Edit" ขั้นตอนที่ซับซ้อนนี้อาจลด Usability ของ Contributor ลง

### Decision:

เลือกให้น้ำหนักกับ Security มากกว่า Usability สำหรับ Contributor เนื่องจาก Contributor มีสิทธิ์ในการเสนอแนะข้อมูลที่ส่งผลต่อคุณภาพของ Dataset ดังนั้น การตรวจสอบสิทธิ์ที่เข้มงวดจึงจำเป็นเพื่อป้องกันการป้อนข้อมูลที่ไม่เหมาะสมเข้าสู่ระบบ แม้ว่าจะทำให้ขั้นตอนการให้ Feedback ลูบง่ายขึ้นเล็กน้อยก็ตาม

