**Subject:** Technical Requirements for Mapillary API Integration - Replacing Google Maps APIs in our Journey Risk Management Platform

**To:** The Mapillary Team  
**From:** [Your Name]  
[Your Designation]  
[Organization Name]  
[Email] | [Phone]  
[Date]

**Dear Mapillary Team,**

We are developing a comprehensive **Journey Risk Management Platform** specifically designed for logistics and transportation safety operations across India. Our platform performs advanced route analysis to identify and mitigate various risk factors including road conditions, sharp turns, blind spots, accident-prone areas, and emergency service availability.

Currently, our platform extensively utilizes **Google Maps Platform APIs** across 13 interconnected services. We are evaluating **Mapillary** as a potential replacement or supplement to achieve better cost optimization, enhanced data coverage, and open-source flexibility.

Below is a detailed breakdown of our current implementation and specific requirements for each service module:

**1. Road Condition Analysis Service**

**Current Implementation:** services/enhancedRoadConditionsService.js

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Road surface quality detection | Google Roads API + Street View | Identify potholes, construction zones, surface deterioration | Need visual detection of road surface conditions from street-level imagery |  |
| Road geometry analysis | Google Roads API | Extract road width, lane count, shoulder presence | Require road attribute data (width, lanes, shoulders) |  |
| Speed limit extraction | Google Roads API Speed Limits | Retrieve posted speed limits for risk calculation | Need speed limit data for route segments |  |
| Road type classification | Google Roads API | Classify as highway/state/district/rural road | Require road classification metadata |  |

**API Call Volume:** ~30 segments per route × 100 routes/day = 3,000 calls/day

**2. Sharp Turns & Blind Spots Detection**

**Current Implementation:** services/sharpTurnsBlindSpotsService.js & services/realBlindSpotCalculations.js

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Elevation profile analysis | Google Elevation API | Detect hill crests causing blind spots | Need elevation data points every 10-20m along route |  |
| Turn angle calculation | GPS geometry + Street View | Calculate turn severity (30°-180°) | Require GPS track analysis or road geometry data |  |
| Visibility distance measurement | Street View + Places API | Assess sight lines around curves/obstacles | Need 360° imagery to analyze visibility obstructions |  |
| Obstruction detection | Google Places API | Identify buildings/vegetation blocking views | Detect structures within 75m of road affecting visibility |  |
| Multi-angle photography | Street View Static API | Capture turn approaches from multiple angles | Need street-level images at 0°, 90°, 180°, 270° headings |  |

**API Call Volume:** ~20 critical points per route × 5 angles = 100 images/route

**3. Emergency Services & Amenities Mapping**

**Current Implementation:** services/dataCollectionService.js (Unified Service)

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Hospital/Police/Fire station location | Google Places Nearby Search | Find emergency services within 50km | Need POI data for emergency services with distance calculation |  |
| Fuel station detection | Places API (gas\_station type) | Locate fuel stops along route | Identify fuel stations with amenities (ATM, restroom) |  |
| Food/Rest stop identification | Places API (restaurant, lodging) | Find driver rest facilities | Detect restaurants, dhabas, hotels along highways |  |
| Service availability hours | Places Details API | Check 24/7 availability | Need operating hours for emergency planning |  |
| Phone numbers extraction | Places Details API | Emergency contact information | Require contact details for critical services |  |

**Service Categories Tracked:** 8 types × 25 segments = 200 searches/route

**4. Accident-Prone Area Analysis**

**Current Implementation:** services/accidentProneAreasService.js

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Historical accident visualization | Street View Timeline | Verify road conditions at accident locations | Need historical imagery to track road evolution |  |
| Intersection complexity analysis | Street View + Roads API | Assess junction visibility and design | Require 360° views of intersections |  |
| Traffic flow observation | Street View imagery | Analyze congestion patterns from visual cues | Detect traffic density from street-level photos |  |

**5. Weather Impact Visualization**

**Current Implementation:** services/enhancedWeatherService.js

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Seasonal road condition changes | Street View (if available for different seasons) | Show monsoon flooding, winter fog impact | Need imagery from different seasons/weather conditions |  |
| Visibility assessment | Street View metadata | Evaluate fog/rain impact on visibility | Extract weather conditions from image metadata |  |

**6. Route Planning & Visualization**

**Current Implementation:** services/apiService.js

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Full route polyline generation | Google Directions API | Create detailed route path with waypoints | Generate route polyline from origin to destination |  |
| Distance & duration calculation | Distance Matrix API | Calculate total journey metrics | Compute accurate road distance (not straight line) |  |
| Alternative route suggestions | Directions API (alternatives=true) | Provide safer route options | Suggest alternate paths avoiding high-risk areas |  |
| Route elevation profile | Elevation API (path mode) | Show altitude changes along route | Elevation data for entire route path |  |

**7. Visual Documentation & Reporting**

**Current Implementation:** Multiple services generate visual reports

| **Feature** | **Google API Used** | **Purpose** | **Technical Requirements** | **Can Mapillary Support?** |
| --- | --- | --- | --- | --- |
| Risk point photography | Street View Static API | Document sharp turns, blind spots in reports | Download high-res images for PDF reports |  |
| Aerial overview maps | Static Maps API (satellite) | Show route context and terrain | Satellite/aerial imagery for route overview |  |
| Route visualization with risk overlays | Static Maps + custom markers | Color-coded risk zones on route map | Ability to overlay risk data on static maps |  |
| Multi-view documentation | Street View (multiple headings) | Comprehensive visual evidence | 360° coverage at critical points |  |

**8. Real-time Data Requirements**

| **Feature** | **Mapillary Alternative Needed** | **Give Details** |
| --- | --- | --- |
| API Rate Limits | What are Mapillary's rate limits? |  |
| Batch Processing | Does Mapillary support batch requests? |  |
| Response Time | Expected latency for image/data retrieval? |  |
| Concurrent Requests | Maximum concurrent connections allowed? |  |

**9. Data Coverage & Quality Requirements**

**Geographic Coverage Needs:**

* **National Highways (NH):** NH-44, NH-48, NH-16, etc.
* **State Highways:** Particularly in Maharashtra, Gujarat, Karnataka, Tamil Nadu
* **Rural/District Roads:** Last-mile connectivity to industrial areas
* **Challenging Terrains:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Image Quality Requirements:**

* **Resolution:** Minimum 2MP for hazard detection
* **Coverage Density:** Images every 20-50 meters in critical areas
* **Recency:** Imagery not older than 2 years
* **Viewing Angles:** Front, side views for turn analysis

**10. Integration & Technical Requirements**

**API Integration Needs:**

* **REST API:** For backend Node.js services
* **JavaScript SDK:** For web-based map visualization
* **Authentication:** API key or OAuth2 support
* **Response Format:** JSON with structured data
* **Error Handling:** Clear error codes and rate limit headers

**Platform Specifications:**

* **Backend:** Node.js with Express.js
* **Database:** MongoDB with Mongoose ODM

**11. Commercial & Licensing Questions**

1. **Commercial Use:** Can Mapillary data be used in a B2B SaaS platform serving logistics companies?
2. **Data Retention:** Can we cache Mapillary images locally for 30 days for offline report generation?
3. **Attribution:** What attribution is required when using Mapillary data in PDF reports?
4. **Private Uploads:** Can we create private organizational sequences for proprietary routes?
5. **API Pricing:** What is the pricing structure for commercial usage exceeding free tier limits?
6. **SLA Guarantees:** What uptime and performance guarantees are provided?
7. **Data Export:** Can we export route analysis results containing Mapillary-derived insights?

**12. Specific Technical Queries**

1. **Vector Tiles:** Does Mapillary provide vector tiles for custom map styling?
2. **Sequence API:** Can we retrieve continuous image sequences along a route?
3. **Object Detection:** Does the API provide detected objects (signs, barriers, vehicles)?
4. **Image Metadata:** What metadata is available (timestamp, weather, camera settings)?
5. **Panoramic Images:** Are 360° panoramic images available for intersection analysis?
6. **Filtering Options:** Can we filter images by date, quality, viewing angle?
7. **Webhook Support:** Can we receive notifications for new imagery in our areas of interest?

**Expected Benefits from Mapillary Integration:**

1. **Community-Driven Data:** Access to crowd-sourced imagery for better rural coverage
2. **Cost Optimization:** Potential reduction in API costs compared to Google
3. **Open Data Philosophy:** Alignment with open-source principles
4. **Frequent Updates:** Community uploads providing more recent imagery
5. **Custom Contributions:** Ability to improve coverage through our own uploads

**Next Steps:**

We would appreciate:

1. **Technical feasibility assessment** for each service category
2. **API documentation** specific to our use cases
3. **Coverage analysis** for Indian highways and rural roads
4. **Proof of Concept** guidance for a pilot implementation
5. **proposal** for enterprise-level usage

Looking forward to exploring how Mapillary can enhance our platform's capabilities while supporting the open mapping community.

**Warm regards,**

**[Your Full Name]**  
**[Your Role], [Your Organization]**  
**Email:** [Your Email]  
**Phone:** [Your Phone Number]  
**LinkedIn:** [Your LinkedIn Profile]