Capstone Project Final Report

Date: August 2, 2025

Team Members: Youyou Huang Aakash Ajay Singh Yash Hulsurkar Vimbai Muyengwa Ashay Koradia Rohan Ghosh

Prepared For: Tier IV

Project Title: Operational Design Domain Safety Visualization tool

Company: Tier IV, Inc

Heinz Corp Organization: Tier IV Capstone Team, Summer 2025 Cohort

Table of Contents

I. EXECUTIVE SUMMARY

Over the course of two months, our team has identified an opportunity to mitigate the issue of automandering, where companies push state-level policies to regions without holistic consideration of safety measures in local environments. The autonomous driving (AD) industry is on the cusp of a revolution, with experts[1] projecting that by 2030, autonomous driving vehicles will have reached level 5 with an over 74 billion market value. Level 5 refers to “fully autonomous vehicles that monitor roadway conditions and perform safety-critical tasks throughout the duration of the trip with or without a driver present” (Center for Sustainability Systems).

As TierIV continues to build its presence in the autonomous driving space, they are seeking to improve upon traditional Operational Domain Design by removing regional and capital constraints through building a visualization tool that solves the mapping problem. An Operational Domain Design or ODD “defines the operating conditions under which an ADS is designed to operate safely” (ISO 34503). In 2022 alone, 42,795 lives were lost in vehicle crashes. 94% of which were attributed to human error. Autonomous vehicles (AVs) present an opportunity to significantly reduce these preventable incidents. By minimizing human error, AVs could reduce crashes by up to 90%, potentially saving an estimated $190 billion annually. TierIV is uniquely positioned to lead this shift, driving a measurable impact on public safety and economic efficiency[2].

Focusing on this issue, the team worked with our Tier IV client partner to understand the limitations of traditional ODD and identify the need for a visualization tool that maps out a road network. With safety of OD at the center, the team identified the following focus areas, outlining findings and implications:

Focus Area	Findings and Implications
Location Agnostic ODD	Specificity of ODD geographic regions presents scaling issues
ISO Standard Adherent:	Data limitations for specified ODD elements limit ISO-34503 guardrails for ODD design

[1] Center for Sustainable Systems, University of Michigan. 2024. "Autonomous Vehicles Factsheet." Pub. No. CSS1

[2] Center for Sustainable Systems, University of Michigan. 2024. "Autonomous Vehicles Factsheet." Pub. No. CSS16-18

ISO Standard Adherent:	Automandering Mitigation: Private company's push for state-level policies after testing in select communities decreases AD safety
Road Networks:	ODD compliance impacts network connectivity for unmapped geographic regions

After conducting workshops and considering the specified timeline, the team determined that the build and prioritization of the ODD elements should be conducted in three phases. The ODD prioritization was informed directly by discussions with our client partner, who guided what elements would be most impactful to address first. Our project work focused on phase one, where we implemented high-priority scenery-related elements such as zones, drivable areas, and parking lots. When a user interacts with our tool, it will sift through over 9.1 billion OSMnx nodes to retrieve relevant Scenery ODD elements. Looking ahead to phases two and three, future project work can expand into more environmental and dynamic elements, including weather, connectivity, traffic agents, and special vehicle types. This phased approach ensures we address the most critical safety context first while setting the foundation for more complex use cases in future releases.

Visualization Tool Current State and Future Work

User interaction with visualization tool sifts through **9.1 billion nodes** via osmnx to retrieve relevant Phase One ODD elements

	Current State	Future Work	
ODD Elements	**Phase One (High Priority)**	**Phase Two**	**Phase Two or Three**
Main Category	Scenery	Environment	Dynamic
Sub Category 1	Zones, Driveable Areas, Junctions	Weather, particulates, illumination, connectivity	Traffic agents, subject vehicle
Sub Category 2	Parking lots, traffic management zone, driveable area signs, driveable area type, roundabout, intersection etc.	Wind, rainfall, natural illumination	Agent type, presence of special vehicles, specification
Sub Category 3	Highway types, Speed limits, Lane dimension	Daytime, nighttime, low ambient	Motor vehicles, ambulances, police vehicle, work vehicle

15

II. PROBLEM STATEMENT[3]

The industry commonly understands that Autonomous Vehicles (AVs) should be designed, developed, and validated with a Specific ODD in mind. An ODD "defines the operating conditions under which an ADS is designed to operate safely" (ISO 34503). Following such safety best practices, AV engineers would need to pre-define an ODD within which the vehicle is intended to operate safely. Then they must design, develop, and validate the system within that ODD before safely deploying the vehicle. Traditional AV companies have defined the ODD as the target deployment region itself, meaning they must conduct their development cycle within those specified physical areas (e.g., if they define San Francisco as the ODD, they must develop, test, and validate within San Francisco).

This approach can lead to severe capital constraints: First, it requires enormous capital investment to physically operate in such massive regions to conduct the development cycle. Second, it can lead to scaling issues when expanding into regions the AV has never driven since the ODD is defined by specific geographic locations. Instead, we want to develop a more lab-like development approach. We would like to define a generalized ODD not tied to a specific geographic region that follows the ISO standards, conduct the development cycle in a location-agnostic manner, and ensure the specified ODD conditions are met throughout the development. By taking this approach, the development can happen anywhere (e.g., simulation environment, closed course test facilities, and public roads that meet the ODD specification, but may not be the actual deployment location, and digital twin recreations of such roads).

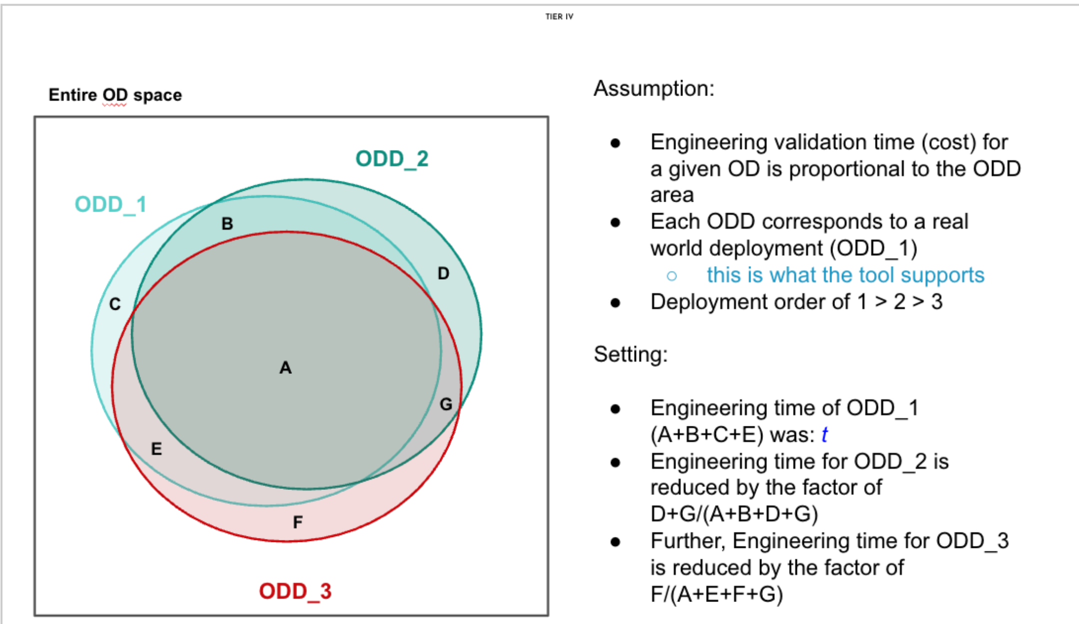
This approach has the possibility of minimizing development costs. Still, it creates another problem that needs to be solved: the mapping problem from the ODD definition to real-world geographic regions. Thus, this project asks students to develop this mapping algorithm and a visual interface that communicates how the change in the ODD will increase/decrease the physical footprint of the deployable region.

[3] Referenced direct client problem statement documentation: Capstone - Operational Design Domain (ODD) mapping tool

III. BUSINESS IMPACT

The mapping problem has led to the creation of a visualization tool to build a “mapping algorithm and visual interface”. At its core, this tool allows us to map specific physical routes to ODD models, enabling clearer decision-making and improved resource allocation.

Key Business Impacts:



1. Engineering Validation Efficiency
   Each route can be modelled as a unique ODD, where validation time scales proportionally with the ODD's spatial footprint. The tool helps quantify which environmental elements (A, E, D, G) have already been validated, reducing redundant efforts when deploying to overlapping or adjacent areas

2. Economies of Scale
   Deployments no longer operate in isolation. Instead, the tool enables knowledge transfer between deployments by identifying elemental and geographic overlaps. For example, if ODD_1 + ODD_2 substantially cover ODD_3, the marginal validation effort (F) is significantly reduced, amplifying cumulative utility across rollouts. Engineering time for ODD_3 is reduced by the factor of F/(A+E+F+G)

3. Deployment Order Optimization
   By understanding overlap, the tool supports intelligent sequencing of test deployments. This allows teams to prioritize routes that unlock the most subsequent coverage, leading to measurable cost and time savings and the elimination of duplicated validation work

4. Strategic Decision Support
   The tool supports business and engineering teams in evaluating deployment scenarios

under business constraints (budget and timeline)

5. Risk Mitigation
   The primary risk mitigation utility of this tool is the fact that with the tool, we can have a reasonable safety argument around safe operational expansion, where, without the tool, that rationale is a little more unclear. So it is more like a 0 to 1 improvement than a 1 to x improvement

IV. Solution

Our project followed a phased, cross-functional approach to build a reusable, data-driven visualization tool for Operational Design Domains (ODDs).

The focus was on three core pillars: data processing, visualization, and strategic problem framing.

1. Data Processing and Backend Enablement

Phased Data Understanding:

We conducted weekly working sessions to assess the data available and define what we could do with it. While data access was straightforward, meaningful data manipulation required careful planning and continuous adaptation around task delegation and system architecture.

Caching and Performance Optimization:

Originally leveraging the OSMnx library to fetch map features online, we implemented a MongoDB caching layer. This allowed us to store previously analyzed features and serve them instantly, improving performance and reducing user wait times.

App Workflow:

The application's architecture[4] is designed for modularity and performance, processing user requests through a four-stage workflow: Inflow, In-Session Cache, Analysis & Storage, and Output.

[4] See next page for architecture diagram

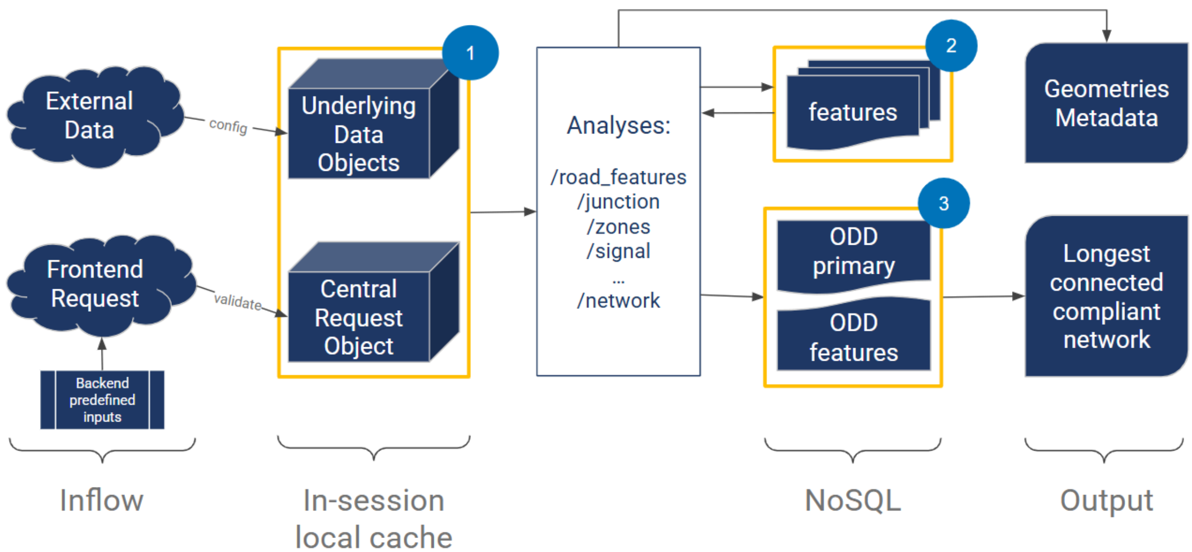


Figure 4.1: Application Workflow Diagram

Stage 1: Inflow:

The process begins with two distinct types of data inflow that serve as inputs to the backend:

- **User-Driven Inflow (Frontend Request & Local Files):** This is the dynamic component where the user interacts with the *React* frontend to define their analysis. At the first window, the user passes geographical location query parameters and basic settings to initialize the current request session.

  .env: User must maintain a local .env file specifying the authentication for the MongoDB NoSQL database:

  - CONNECTION_STR
  - USERNAME
  - PASSWORD

  Backend predefined inputs: A file with predefined inputs, currently in *"/predefined/user_predefined_inputs.xlsx"* of the backend project directory, including the default query and a list of ODD components, is parsed and saved at the start of the current session. While these inputs are not dynamically manipulatable by the user in-session, they are designed to serve as an anchor point for analyses and are only manipulatable by personnel responsible for backend maintenance.

- **System-Driven Inflow (External Data & Configuration):** This is the static component where the backend ingests the foundational data required for any analysis. This includes pulling the raw road network graph from the OSMnx library and loading any central configuration and supplementary proprietary data.

Stage 2: In-Session Local Cache:

To optimize performance, the backend maintains a simple in-session cache.

Underlying Data Objects: Configured geospatial data queried from OSMnx is stored temporarily to avoid redundant API calls and maintain consistency for the same geographic area within a session used across all analysis pipelines.

Central Request Object: The user's validated request from the frontend and predefined inputs are combined with the relevant geospatial data to create a single, unified "Central Request Object." This object serves as the single source of truth for feature analysis pipelines, ensuring consistency and preventing parsing disparities.

Stage 3, 4 & 5: Analysis & Storage (NoSQL) & Outputs:

The **Central Request Object** and accompanying **Underlying Data Objects** are passed to the core analysis engine, which performs a series of modular evaluations. These assessments analyze feature extracts through both metadata and geometric criteria, focusing on **Road Feature**, **Junctions**, **School Zones, Parking Lots**, and **Traffic Signals**.

Feature Analysis: Each feature domain undergoes targeted extraction and geometric processing. The system returns structured outputs containing all relevant geometries, metadata, and a feature dictionary specific to each analytical pipeline.

- **Road features**
  This module analyzes individual road segments, extracting metadata such as lane counts, presence, lane marking presence and type, direction of travel, and posted speed limits. These insights are essential for assessing whether a road's physical structure meets the operational criteria of autonomous driving systems.
- **Junctions**
  - *Junction type*: Identifies and classifies all junction types within the network, including crossroads, T-junctions, Y-junctions, roundabouts, and more.
  - *Conflict counter*: Quantifies potential conflict points based on permitted turning movements and predefined edge-pair conflict criteria.
  - *Polygon Representation*: Constructs visual polygons representing each junction's complexity by aggregating short segments of adjacent edges (with configurable range), enabling clear frontend visualization.
- **Signals**
  This module focuses on detecting regulatory elements that influence traffic flow, particularly the presence of traffic lights at junctions. This information is crucial for ensuring autonomous systems conform to traffic laws and safely manage signalized intersections.

- **School Zones**
  This module identifies educational facilities using geospatial data and generates buffer zones (default radius: 100 meters, configurable). Overlapping zones are merged into a unified region, with aggregated metadata treating complex school zones as single operational entities.
- **Parking Lot**
  The module processes geospatial points representing parking facilities and converts them into polygonal zones using buffers (default: 15 meters, configurable). Overlapping areas are merged and standardized using 2-meter buffer offsets to define consistent approach and departure corridors.

Feature Map: Alongside baseline analytics, the system provides a comprehensive feature map detailing all possible attribute values for each pipeline. This map enables dynamic ODD selection on the frontend, facilitating flexible network analysis and minimizing the need for hard-coded configurations or ongoing maintenance.

Network Analysis: The engine identifies the longest connected subnetwork that complies with selected ODD criteria. Users may choose one of the following modes:

- *All*: Displays the longest connected road network within the queried boundary, irrespective of feature metadata.
- *Predefined*: Filters the network according to ODD specifications provided via the user-defined Excel input.
- *Live*: Dynamically filters the network based on the user’s session selections and results from the feature analyses.

NoSQL Storage:

The system leverages MongoDB for persistent storage of analysis results and network data, optimizing both retrieval and resource efficiency.

- **Feature-specific collections**

  Long-running analysis outputs (e.g., junction metadata) are cached in dedicated MongoDB collections to prevent redundant computation and accelerate access.

- **Network collections**
  - *network_primary*: Stores edge-level road feature documents. This serves as the primary dataset for ODD-compliant "most connected road network" analysis.
  - *network_feature*: Contains metadata for node-level features, acting as the secondary input for the same analytical pipeline.

List of core tools and resources:

Name	Library/Source	Domain	Usage Note
OSMnx	osmnx	Geospatial/Networks	Street network analysis, graph operations
GeoPandas	geopandas	Geospatial	Spatial data manipulation, geometric operations
NetworkX	networkx	Graph Analysis	Graph data structures for network topology
Flask	flask	Web Framework	API endpoints, blueprints
MongoDB	pymongo	Database	Bulk operations, data persistence
React Native	react-native	Frontend	Frontend Development

2. Frontend Visualization and User Experience

- Framework Selection: We selected React for its modern library support and compatibility with map visualizations, outperforming other options like Angular and Vue.js in terms of flexibility.

- Interactive Data Parsing: The frontend was designed to parse complex, multi-attribute data sets and deliver a highly interactive user experience. Visualization plays a critical role not just in usability, but also in investor engagement. The ability to explore cities and routes visually creates immediate and memorable impressions essential for storytelling and fundraising.

3. Problem Framing and Iterative Design

- Reevaluating the Problem Statement: We revisited the original assumptions, such as prioritizing the longest route, and instead explored alternative definitions of route importance (e.g., number of nodes or areas covered).

- Lessons from Prior Iterations: The previous version of the tool output only ODD elements, which limited its usefulness. Learning from this, we adopted a more iterative and modular approach, focusing on value-added outputs aligned with business and technical goals.

- Algorithm Development for Zones: A key deliverable was an adaptable algorithm to interpret and extract zoning information from open-source mapping tools, tailored to user input and deployable to the frontend.

V. DATA ANALYSIS

To demonstrate the practical utility of ODDScope, we conducted a sample analysis for the city of Shiojiri, Nagano, Japan.

Scenario: Tier IV wishes to evaluate Shiojiri for a potential deployment of an AV designed for arterial roads.

1. Defining the ODD Parameters: The user defines the following ODD via the tool's interface:

A. Traffic signals
B. School zones
C. T Junctions
D. Lanes

2. Data Processing and Analysis:

- ODD scope sends a request to the backend for the road network of Shiojiri.
- The backend queries OpenDrive via OSMnx, retrieving all road segments and their associated metadata (road type, speed limits, lane counts, surface type, etc.).
- The analysis engine iterates through every road segment, filtering out any that do not comply with the defined ODD. For example, a residential street is discarded, as is a primary road with a speed limit of 80 km/h.
- The remaining compliant segments form a new, "safe" road network graph.

3. Visualization and Strategic Insight: The frontend renders the results on an interactive map:

- **Compliant Roads:** All roads meeting the ODD criteria are highlighted.
- **Non-Compliant Roads:** All other roads are greyed out, providing immediate visual context of the operational boundaries.
- **Longest Connected Route:** The tool runs a network analysis algorithm on the highlighted segments and highlights the single longest path the vehicle could travel without leaving the ODD in blue.

VII. LESSONS LEARNED

The project provided valuable insights across technical, project management, and strategic domains.

- **Technical Lessons:** The Criticality of Caching: Our initial architecture with direct, real-time calls to the OSMnx API resulted in significant performance bottlenecks.

Implementing the in-memory caching layer was the single most important technical decision we made, transforming the tool from a slow prototype into a responsive application.

- **Team Coordination:** As a six-person team, we learned that effective coordination required more than just technical skills and demanded clear communication and structured collaboration frameworks. Managing team members across backend development and frontend taught us the critical importance of sprint planning with well-defined tasks and sequencing to prevent blocking dependencies.
- **Client Relations:** Our team established clear and consistent communications with our client Tier IV, through regular progress meetings where we demonstrated development milestones and collaboratively discussed emerging ideas. Given our limited project timeframe, we worked closely with the client to filter and prioritize which ODD zones to focus on based on their importance to autonomous vehicle safety and operations. This collaborative approach to requirement prioritization ensured our development efforts aligned with the client's most critical needs while effectively managing project scope within our constraints.
- **Documentation:** Through our collaborative development process, we recognized that comprehensive documentation served as the backbone of effective teamwork and successful client handoff. Creating and maintaining technical documentation, user guides, API specifications, and system architecture diagrams became a shared responsibility that facilitated smoother collaboration and knowledge transfer.
- **Visual Storytelling:** Our collaborative work on demo strategy development and video production taught us valuable lessons in technical storytelling and visual communication of complex software capabilities. Creating the project trailer as a team effort enhanced our ability to distill technical achievements into compelling narratives accessible to non-technical audiences. This was also a great opportunity to use AI and learn its video capabilities.

VIII. SUGGESTIONS FOR FUTURE WORK

ODD scope provides a robust foundation that can be extended in several high-value directions. We recommend Tier IV consider the following enhancements:

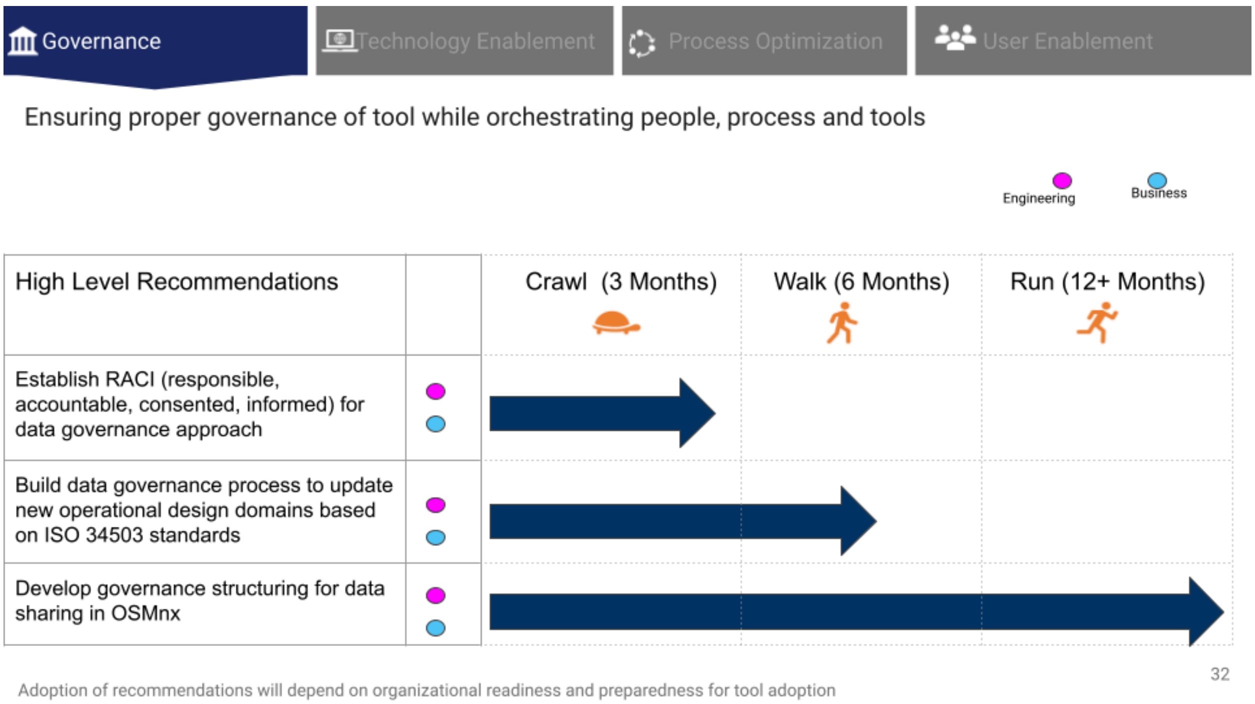
- **Data fusion between public and proprietary data:** The primary scenery data source, OpenStreetMap (OSM), exhibits coverage gaps in certain regions. To complement this, TierIV's proprietary dataset, comprising temporal snapshots of road conditions captured from moving vehicles, was considered. Due to time constraints, full integration of the proprietary data into the framework was deferred. However, we designed preliminary utility functions as reference scaffolds to support future data mapping and ongoing development.

- **Integration of Dynamic Data Sources:** Enhance the ODD model by incorporating real-time data via APIs. This could include:
    - **Weather Data:** Dynamically restrict the ODD based on current conditions (e.g., disabling operation in heavy rain or snow).
    - **Traffic Data:** Adjust routing and operational permissions based on live traffic congestion or road closures.
- **Sophisticated Risk Assessment Model:** Evolve the current binary compliance model (a road is either in or out) to a more granular, weighted risk scoring system. This would allow segments to be scored on a spectrum of safety, enabling more nuanced decisions (e.g., "low risk," "medium risk," "high risk").
- **Cost-Benefit Analysis Module:** Develop a feature that allows users to model the financial trade-offs of different deployment strategies. For example, the tool could estimate the engineering cost required to add a new feature (e.g., handling roundabouts) and project the corresponding increase in operational network coverage and market reach.
- **Expansion of Data Sources:** Integrate proprietary or municipal datasets (e.g., official accident data, detailed infrastructure surveys) to increase the fidelity and accuracy of the analysis beyond what is available in open-source data alone.
- **UI/UX Enhancements:** Add features for saving, comparing, and sharing different ODD scenarios. This would transform the tool into a fully-fledged collaborative platform for strategic planning.
- **Reusable Infrastructure:** To enhance scalability and user experience, the second phase of visualization tool implementation should extend the reusable infrastructure by leveraging Docker containerization for seamless deployment across local and cloud environments. Future iterations will introduce support for multiple users, each with their isolated profile and workspace. Additionally, we aim to enable distributed and parallel execution capabilities to improve performance and facilitate collaborative work on the tool across multiple nodes or machines.

IX. CONCLUSION/ RECOMMENDATIONS

Looking ahead to ensure the success of the phase one implementation of the visualization tool for Tier IV, we surfaced high-level recommendations for tool adoption. These high-level recommendations encompass governance, technology enablement, process optimization, and user enablement. Adoption of our recommendations will be dependent on organizational readiness and preparedness. The following outlines these high-level recommendations with consideration of the end user in business or engineering. Our recommendations have been outlined in a crawl, walk, run format, providing an estimated timeline that can be adjusted according to the allowable pace and appropriate sequencing for Tier IV tool adoption.

Governance: The framework of roles and responsibilities, and data governance structures



The governance-focused recommendations are to support the successful adoption of a visualization tool for Tier IV, structured in a crawl (3 months), walk (6 months), run (12+ months) timeline. The objective is to ensure proper governance of the tool while orchestrating people, process, and tools. Each recommendation involves collaboration across engineering and business functions. Implementation is staged and progressive, based on organizational readiness

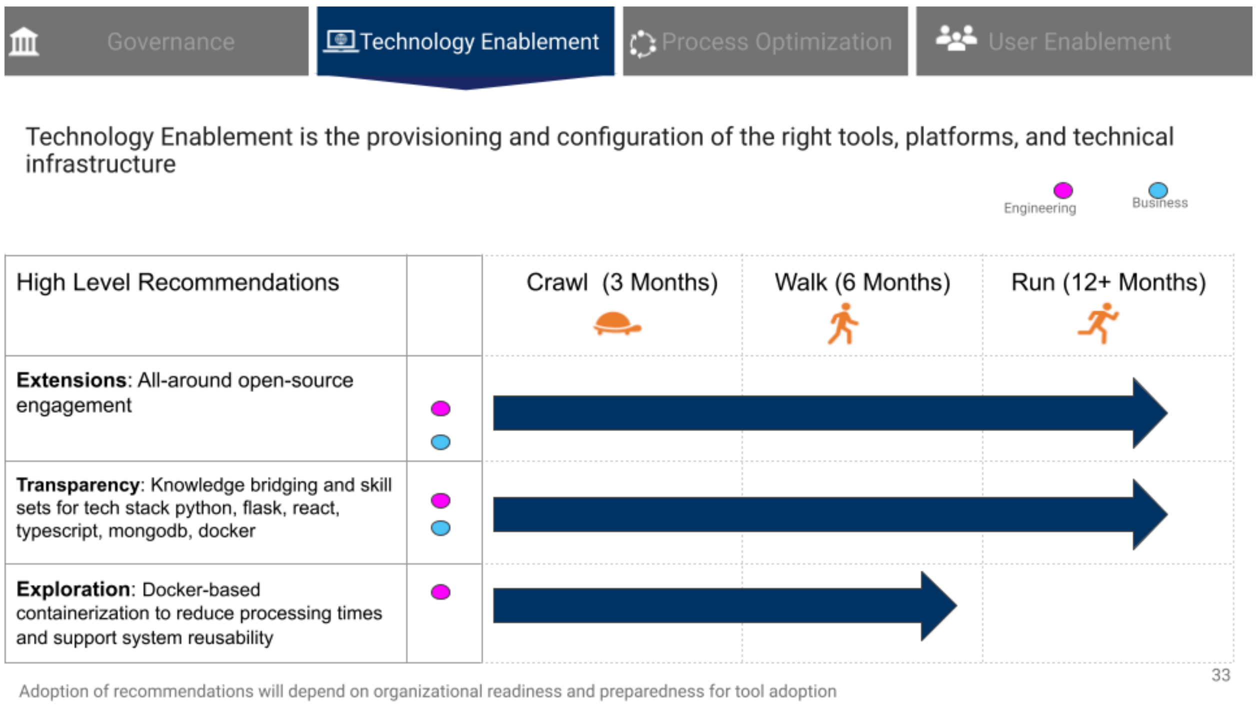
High-Level Recommendations:

1. Establish RACI (Responsible, Accountable, Consented, Informed) for the data governance approach. Involves both engineering and business roles.Begins in the crawl phase.

2. Build a data governance process aligned with ISO 34503 standards to update new Operational Design Domains (ODDs). This recommendation is a joint responsibility of engineering and business, and starts in the crawl stage and continues into the walk.

3. Develop governance structuring for data sharing using OSMnx. This recommendation is a combined engineering-business effort. Starts at crawl and continues through walk into run.

Technology Enablement:



Objective: Provision and configure the right tools, platforms, and technical infrastructure to support the visualization tool.

High-Level Recommendations:

1. Extensions – *All-around open-source engagement.* Involves both engineering and business. Begins in crawl and continues through run phases. Aligns with TierIV's mission and its commitment “to the formation of open communities, in which we take the initiative to make it possible with our partners to achieve our transformational vision.”[5]

2. Transparency – Knowledge bridging and skill development for the tech stack (Python, Flask, React, TypeScript, MongoDB, Docker). Requires collaboration between engineering and business to create the appropriate technical support channels and ensure the availability of necessary skillsets in the visualization tool implementation. Spans from

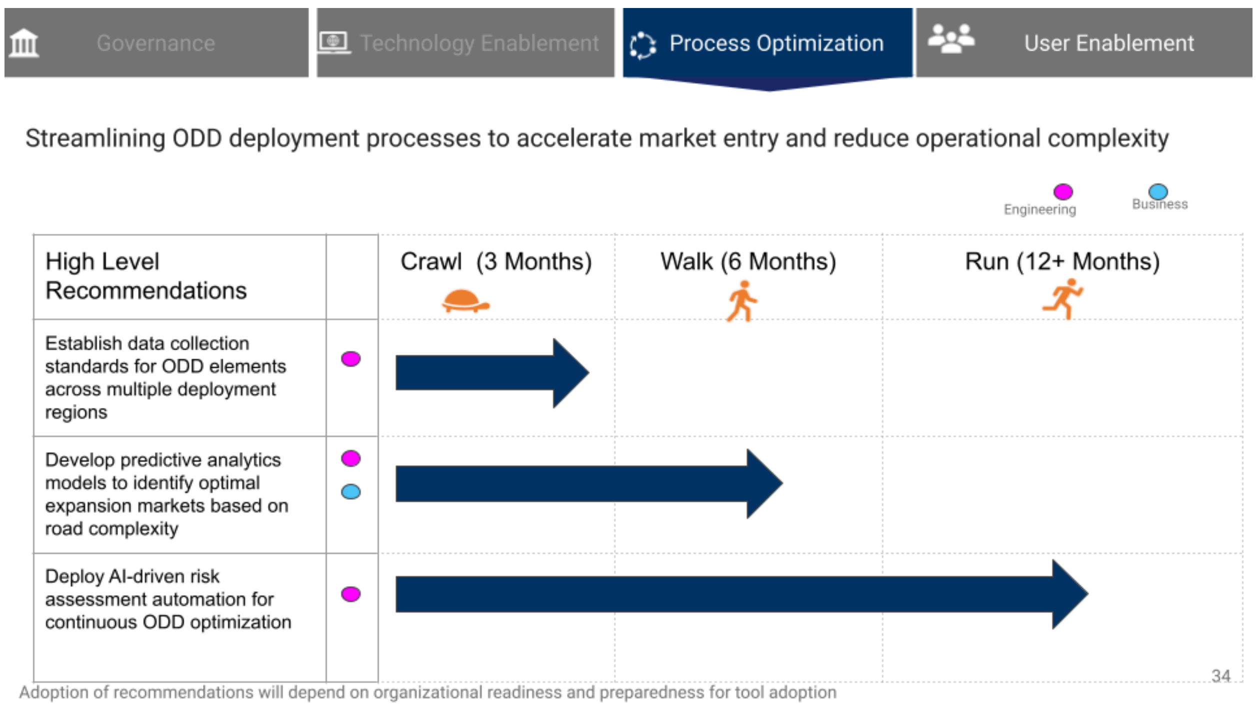
[5] https://tier4.jp/en/about/

crawl through run.

3. Exploration – *Docker-based containerization* to reduce processing times and support system reusability. Primarily driven by engineering. This is a future-looking recommendation whose ideation began in the crawl phase, but will see full maturity in the walk phase.

In summary, emphasis is placed on open-source, skill alignment, and scalable technical solutions. All efforts are aligned with a phased approach and organizational readiness. Engineering leads implementation, with business involvement where cross-functional skills and transparency are needed.

Process Optimization:



Objective: Streamline ODD (Operational Design Domain) deployment processes to accelerate market entry and reduce operational complexity.

High-Level Recommendations:

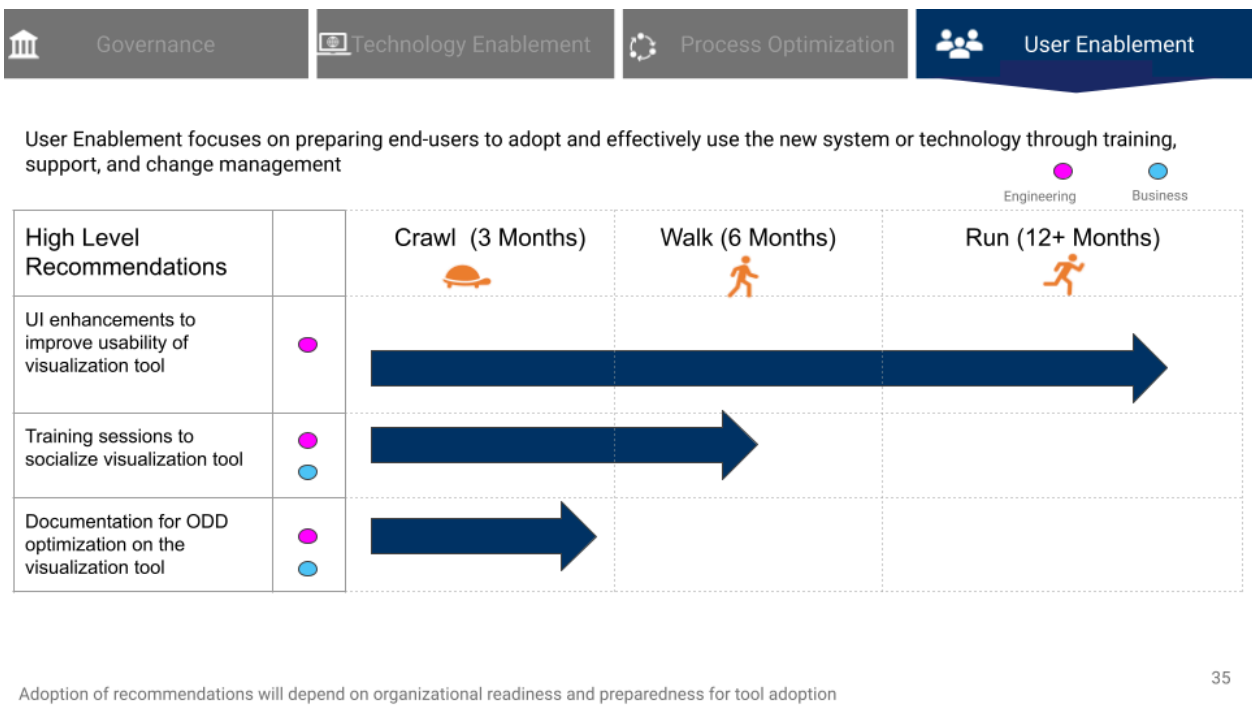
1. Establish data collection standards for ODD elements across multiple deployment regions. This area is driven by engineering. Starts in the crawl phase and concludes in the walk phase

2. Develop predictive analytics models to identify optimal expansion markets based on road complexity. Joint effort between engineering and business. Begins in crawl, extends into walk stages.

3. Deploy AI-driven risk assessment automation for continuous ODD optimization. Led by engineering. Starts in crawl and continues through walk and run

In summary, emphasis is on standardization, data-driven decision-making, and automated risk management. Engineering plays a central role, while business is brought in for strategic market identification. Recommendations are phased, with increasing complexity and value from crawl to run. Implementation depends on Tier IV's organizational readiness and technical maturity.

User Enablement:



Objective: Prepare end-users to adopt and effectively use the visualization tool through training, support, and change management.

High-Level Recommendations:

1. UI enhancements to improve the usability of the visualization tool. Led by engineering begins in crawl and continues through run. This is a continuous improvement initiative that requires a test-driving tool in a real-world setting with business and engineering to

find ways to improve the tool

2. Training sessions to socialize the visualization tool. Involves both engineering and business. Starts in crawl, continues into walk, and is applied on an ad hoc basis depending on the onboarding of the business and engineering team cadence

3. Documentation for ODD optimization on the visualization tool. Joint responsibility of engineering and business. Implemented during the crawl phase

In summary, User enablement begins early and is foundational to adoption success. Focus areas include interface usability, user training, and technical documentation. Business stakeholders are included to ensure tool familiarity and cross-functional alignment. Engineering leads implementation, especially on the UI and documentation fronts. Timeline is sequenced to gradually increase user capability and engagement.

Works Cited:

Center for Sustainable Systems, University of Michigan. 2024. "Autonomous Vehicles Factsheet." Pub. No. CSS16-18

Note on AI Assistance:

This report was written with original ideas and analysis by the project team. ChatGPT, an AI tool, was used to assist with language refinement, formatting clarity, and summarization. All core content, findings, and recommendations are the team’s own. No proprietary or confidential client information was used inappropriately or shared with the system