# Project Minard: A Platform for War-Rooming and Geospatial Analysis in Virtual Space

Stephen A. Bailey, Justin R. Renga, Joseph T. Downs, Miller L. Wilt, Brock A. Wester, and Jordan K. Matelsky

AR VR MR

## ABSTRACT

*The geographic movement of individuals and assets is a complicated maze of relational data, further complicated by the individuals' relationships or allegiances to organizations and regions. Understanding this depth and complexity of information is difficult even on purpose-built systems using conventional compute architectures. A Johns Hopkins University Applied Physics Laboratory (APL) project, called Minard, upgrades the war room to a virtual reality (VR) space. This system provides analysts with a collaborative and secure virtual environment in which they can interact with and study complex and noisy data such as alliances, the transit of individuals or groups through 3-D space, and the evolution of relationships through time. APL engineers designed intelligent visualization systems to bring the best of human intuition to state-of-the-art VR, with human–machine teams interacting both through the VR headset and behind a conventional computer terminal.*

## BACKGROUND

It is interesting to look back through the historical record at some of the earliest existing written maps to see what details their creators thought were important enough to memorialize in stone. One of the earliest examples of a map—if it can be called one—is the Babylonian *Map of the World* (Figure 1), a stone tablet now on display at the British Museum in London.[1] This map displays Babylon at the center of a circle, surrounded by seven mountains. But unlike modern maps, the regions carved into this map do not represent geographic regions: instead, they represent tribes, communities, or notable landmarks. Slightly more modern Inuit cartography devices, such as the Ammassalik wooden oceangoing maps, likewise ignore explicit distances, in this case favoring the portrayal of relevant coastline and island details through simple carved wooden sculpture.[2]

While these examples may seem exotic, other examples might be more familiar to the modern mind: the "war room" portrayed in the film *Doctor Strangelove* displays the USSR in silhouette, with only notable landmarks labeled (Figure 2). In place of the usual detail found on a map, this map instead renders the trajectories of nuclear-armed bomber planes. These maps—and many others like them, spanning thousands of years of history—have one thing in common: the story they tell is of the traversal and utilization of space, and irrelevant details are omitted entirely.

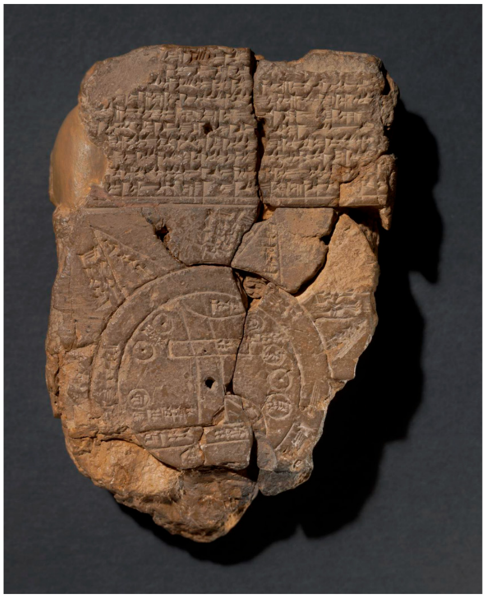


**Figure 1.** An early Babylonian *Map of the World*, c. 6th century BC. Much like its contemporaneous maps, the map (circle graphic with triangular motifs) points to tribes and major landmarks rather than geographically precise locations. (Image courtesy of the British Museum, London, United Kingdom.[1])

By 1869, a cartographer named Charles Joseph Minard leveraged this abundant historical inspiration to create one of the first deliberate marriages of cartography and data visualization for public consumption. Minard's "figurative map," shown in Figure 3, delivered large amounts of relevant information to the viewer quickly and effectively. Note the colocation of quantities, such as army size, with geographic information such as the location of Moscow and the paths of rivers. This chart is now hailed as a victory of data visualization—a case where high informational density is decoupled from visual clutter.[3]

## OVERVIEW OF PROJECT MINARD

Minard was constrained by what a modern artist would now consider a cripplingly limited tool kit. Our team proposes Project Minard, a modern-day extension of the Minardian data visualization philosophy. This work is designed to combine the benefits of modern-day cartographic and geographic information system (GIS)-based geospatial rendering with high-quality 3-D data visualization. This tool enables analysts and researchers to explore spatially distributed data in its native multidimensional and multimodal form.

For example, rail, air, or road traffic may be visualized along the transit corridors represented in the data, or individual units might be tracked across a map. The movement of individuals, sales, or resources may be tracked with a bird's-eye view across the globe. Furthermore, our tools are designed to also enable real-time data

**Figure 2.** The war room as depicted in the film *Doctor Strangelove*. The maps on the wall convey virtually no usable geographic information beyond that required for the relevant tasks. (Public domain image from Ref. 4.)

exploration. Analyses may be performed on these data by interacting in a virtual space, and the effects can be viewed by users in real time, much like how participants in a war room might manipulate pieces on a table-size battlefield. For example, an air traffic control operator might choose to watch the impact of temporarily shutting down a runway; a hospital administrator may want to explore the implications of moving a population of patients from one region to another; or perhaps an epidemiologist might explore how viruses transit international borders.

Key to this process is minimizing the time required to close the feedback loop between a user's input and the system rendering the input's downstream effects. To conduct this research collaboratively, our software uniquely includes the integration of multiple users' inputs in the same virtual space.

While virtual reality (VR) may not afford a user the same level of precision as professional analysis tools or software libraries do, it is our intention that this system will enable the formation of immediate visual intuitions where they were not previously accessible.

## ARCHITECTURE

Among his many works, Charles Minard constructed highly informative maps of rail traffic. Whereas the French railroad in Minard's day stretched only around 9,000 rarely traveled miles and transported only 102 million ton-kilometers per year, the same graphic today would need to cover more than 25,000 miles of rail and more than 30 billion ton-kilometers per year.[5,6]

To accommodate such immense data sets, our approach combines a VR immersive platform for rendering outputs with server-based data analytics software for data analysis. These systems communicate through a generalizable application programming interface (API), which enables highly flexible transit of data sets for visualization in overlay on a geospatial map.

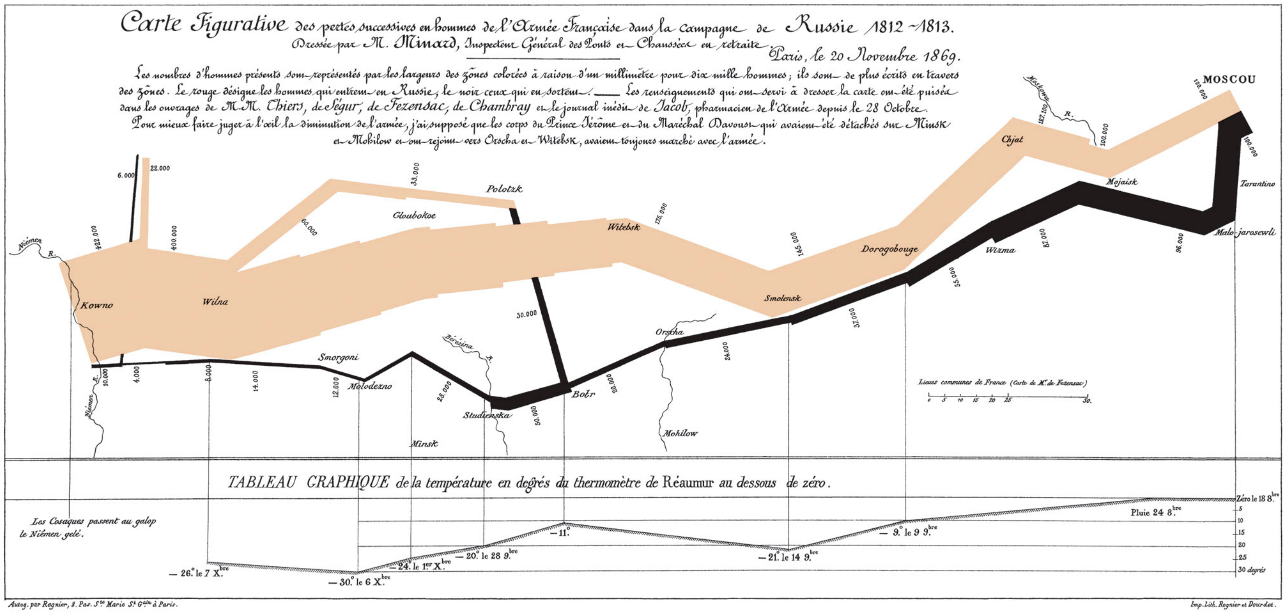



**Figure 3.** Charles Minard's *carte figurative*, or "figurative map," of Napoleon's doomed invasion of Russia in 1812. Reproduced here in its original French, the map conveys geospatial position, temperature and weather, and the scale of the diminishing French Grande Armée. (Public domain image from Ref. 7.)

## Architecture at Scale

The Minard system is composed of a database, which is encrypted at rest and stores original data sets alongside user-defined modifications; a back-end service, which acts both as database manager and as an authentication and analysis platform; an arbitrary number of front-end visualization endpoints, including (but not limited to) VR headsets; and an asset datastore, which stores and publishes nontabular formatted data (such as 3-D objects, point clouds, or other multimedia) to the visualization endpoints. See Figure 4 for an illustration.

## Render Abstraction

The Minard server exposes an API against which a front-end rendering platform may authenticate. Once

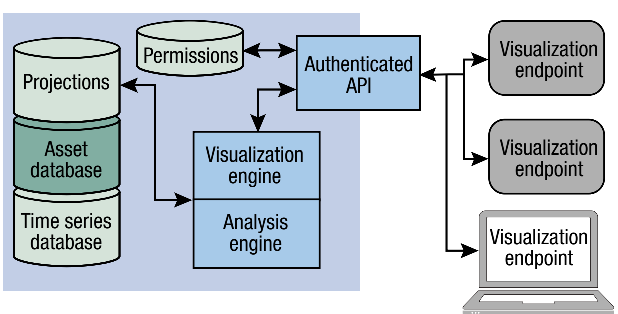



**Figure 4.** An illustration of the Project Minard architecture. On the right, several visualization endpoints (such as VR headsets or a laptop) receive data from the server-side API, which transmits data queried from a database based on the permissions of the user requesting the data.

authenticated, the data transmitted from the server do not contain rendering instructions, and the presentation of these data to the end user is the responsibility of the visualization terminal. This means that a user may attach a VR headset and work along another user who is perhaps interacting directly with the data through a conventional laptop or workstation. Our implementations of Python client-side library and Unity-based visualization endpoint are two examples of Minard API consumers.

## MinardVR: VR Visualization Endpoint

We have implemented an example visualization endpoint for VR headsets using the Unity engine, selected because of its flexibility in targeting a wide variety of consumer VR and mixed reality headsets, including the Oculus Rift, the Microsoft HoloLens, the HTC VIVE, and other non-headset hardware.[8]

The VR platform undergoes a series of life cycle events as the system comes online, is used by an end user, and potentially makes changes to the underlying datastore. This life cycle begins with loader components retrieving data from datastores such as the Minard API and an asset datastore. Loaders then hand off ingested resources to a series of specialized managers. These loaders and managers approximate a Model-View-Controller (MVC) architecture.

Loaders are responsible for ingesting data from the database (via the Minard server API) or from an asset manager. Metadata and information about the entities that inhabit the map (for example, individuals, planes, or subway cars) are loaded from the Minard API. Scene

**Table 1.** VR visualization endpoint loaders and managers

| Name | Responsibilities |
|---|---|
| **Loader** | |
| Map loader | Control the loading of 3-D or 2-D map resources on which the rest of the simulation will be performed. |
| Entity loader | Load "entities"—or atomic components of the simulation (such as human individuals, objects, vehicles, or other tangible assets) from the Minard API server. |
| Location loader | Load static regions and locations (such as state borders or the coordinates of rest stops along a highway). |
| Entity path loader | Load the dynamic locations and movement plans of entities from the API server. |
| Media-entity association loader | Correlate media from the asset server with the appropriate entity or location, using metadata from the API server. |
| Relationships loader | Construct a directed graph between all entities and all locations, where edges represent real-world relationships such as familial ties (X "Is Father To" Y), associations (X "Lives At" Y), or affiliations (X "Works For" Y). |
| Events loader | Ingest event objects from the Minard API into the simulation timeline. |
| **Manager** | |
| Simulation time step manager | Control the passage and speed of simulation time. |
| Entity lifetime manager | Control the entry or exit of entities from the scene as they are created or destroyed. |
| Relationship manager | Manage the directed graph produced by the relationships loader. |
| User selection manager | Convert user selection and interaction actions into rendered outputs or database manipulations. |
| Events manager | Control the occurrence of discrete events and correlate with the clock information from the simulation time step manager. |
| Media-Entity association manager | Handle the dynamic rendering of (potentially large) multimedia files for the user and accept newly loaded material from the media-entity association loader. |

objects like a map, multimedia, and 3-D models are loaded from the asset manager.

Once resources are loaded into the Unity scene, managers are responsible for manipulating and updating the state of the objects as a simulation is interactively animated. For example, a time step manager maintains simulation time and ensures that a reliable "clock" signal is available to all other managers. An event manager keeps track of events taking place, such as the temporary closure of a shipping port or the meeting of two heads of state at an embassy. Complete lists of managers and loaders are available in Table 1.

Our implementation takes inspiration from state-management patterns such as Flux, decoupling render cycle from state management.[9] For example, as entities traverse the map during a simulation, they are "drawn" to their next destination by a finite force that is scaled in proportion to the distance the entity must cover before the next time step (as dictated by the simulation time step manager). Hearkening to historical figurative maps, entity positions are only as specific as the resolution and data density the map allows. In many conventional visualization strategies, multiple entities can inhabit the same 3-D position in space (i.e., they do not collide). While this precise localization mode may be more faithful to the measurement, it is nearly useless when presented to a human viewer, as the visualization becomes unintelligible.

Instead, in our Minardian visualization, our entities "trend" toward their true location but can collide with each other to form clusters of individuals (Figure 5). These moving and shifting clusters are far more visually informative than they would have been had the positions been reported as precisely as possible. Additionally, this design decision enables the VR engine to gracefully handle more moving entities, as the exact location of entities becomes the responsibility of the engine and does not need to be tightly correlated with the database at every rendered frame.

## Minard API Server

To present the necessary volume of data through an authenticated API, we developed the Minard API server in Python, using the Flask HTTP server framework.[10]

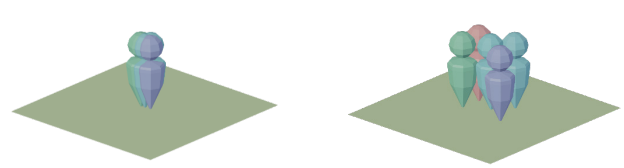


**Figure 5.** Minardian visualization. Many individuals crowd around a point on a small square map. On the left, "precise" entity placement obscures the number of entities in the scene and reduces the clarity of the visualization. On the right, colliding entities better illustrate the salient features of the data set.

The Python programming language provided us with the necessary combination of both well-supported analysis libraries, including NumPy, SciPy, and SciKit for numerical computation and NetworkX for graph analysis, as well as robust networking libraries for transferring data to and from a database and the front-end visualization endpoints.[11–13]

Our implementation of the Minard API server exposes a stateless JSON API to the visualization clients. Depending on the use case, this API may be authenticated, and it is our intention to enable differential authorized access to various data elements from the database. In this way, it will be possible for multiple users to engage with the same virtual scene without sharing private or sensitive information with all users equally.

## CONCLUSION

A generalizable framework for geospatial embedding of data visualization products has far-reaching applications in a variety of domains, including health care, defense, and civil engineering. Our proposed solution to this open question, Project Minard, addresses many of the shared needs of these domains: tracking individual actors or assets through time and space; interrogating the relationships between those actors and assets and how they change over time; and revealing how perturbations to these data sets and the consequences thereof can inform decision-making or strategy. Our software has been developed with extensibility and flexibility in mind, and it is our intention that Minard's software capabilities will scale gracefully with the needs of its user base.

Particular target candidates for further study include disaster recovery and response planning, transport (freight and public) control, and the monitoring of health information both at a local (building-wide) scale as well as at an international scale. It is our hope that these software patterns and architectures will guide future data visualization efforts and will enable higher-quality, more intuitive interpretations of complex data sets across many domains of research.

## REFERENCES


1"The Map of the World." British Museum. https://www.britishmuseum.org/collection/object/W_1882-0714-509 (accessed Feb. 1, 2019).

2A. M. MacEachren, "A linear view of the world: Strip maps as a unique form of cartographic representation," *Am. Cartogr.*, vol. 13, no. 1, pp. 7–26, Jan. 1986.

3C. J. Minard, "Tableaux graphiques et cartes figuratives . . .," Bibliothèque numérique patrimoniale des ponts et chaussées, 1869.

4S. Kubrick (dir.), *Doctor Strangelove*, distributed by Columbia Pictures, Jan. 1964, public domain image from *Wikimedia Commons*, https://commons.wikimedia.org/wiki/File:Dr._Strangelove_-_The_War_Room.png.

5A. Mitchell, *The Great Train Race: Railways and the Franco-German Rivalry, 1815–1914*, 1st ed. New York: Berghahn Books, 2000.

6"Railways, passengers carried (million passenger-km) – France." World Bank. https://data.worldbank.org/indicator/IS.RRS.PASG.KM?locations=FR (accessed Feb. 1, 2019).

7C. J. Minard (1781–1870), public domain image from *Wikimedia Commons*, https://commons.wikimedia.org/wiki/File:Minard.png.

8"Unity." https://unity3d.com/ (accessed Feb. 1, 2019).

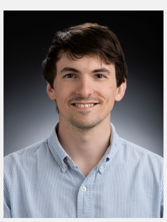
9"Flux." https://facebook.github.io/flux/ (accessed Feb. 1, 2019].

10"Flask—Flask 1.0.2 documentation." http://flask.pocoo.org/docs/1.0/# (accessed Feb. 1, 2019).

11T. E. Oliphant, *Guide to NumPy*, 2nd ed. US: CreateSpace Independent Publishing Platform, 2015.

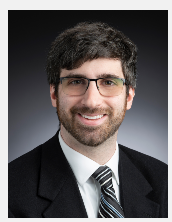
12E. Jones, T. Oliphant, P. Peterson, and others, *SciPy: Open source scientific tools for Python*, 2001.

13A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python in Science Conf.*, Pasadena, CA, 2008, pp. 11–15.

**Stephen A. Bailey,** Air and Missile Defense Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Stephen A. Bailey is a software engineer in APL's Air and Missile Defense Sector. He has a BS in premedical studies from Penn State and an MS in computer science from Johns Hopkins University. He has a diverse set of skills including full-stack web applications development, virtual/mixed reality devices and development, and Unity game engine development. He has worked as the principal investigator for multiple independent research and development (IRAD) projects at APL and serves on an APL committee that funds innovative new ideas across the Air and Missile Defense Sector. His email address is stephen.bailey@jhuapl.edu.



**Justin R. Renga,** Air and Missile Defense Sector, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

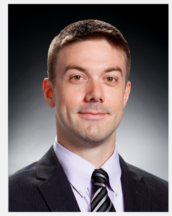
Justin R. Renga is a member of the Command, Control, and Communications Systems Group in APL's Air and Missile Defense Sector. He holds a BS in games and simulation arts and sciences and computer science from Rensselaer Polytechnic Institute and an MS in computer science from Johns Hopkins University. His background is in software engineering, computer science, game development, and user interface design. He has worked on various computer science and software engineering projects while at APL, including researching a method to perform in-line detection and correction of abnormal C++ program behavior. Before joining APL, Justin worked at Sotera Defense Solutions as a defense contractor assisting in the development and maintenance of mission-critical software. His email address is justin.renga@jhuapl.edu.

**Joseph T. Downs,** Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Joseph T. Downs is a data scientist in APL's Intelligent Systems Branch. He received his bachelor of science degrees from the University of Maryland, College Park in physics, mathematics, and computer science, and he completed a master of information and data science degree at the University of California, Berkeley. In his work, Joe has brought computational and statistical methods to bear on problem domains ranging from genomics to space mission planning to neuroscience to national defense. He recently led development of an analysis pipeline for FEMA's national response to COVID-19 on a White House task force, and he has supported image processing and network analysis of data for the Intelligence Advanced Research Projects Activity MICrONS project on petascale neuroscience. His email address is joe.downs@jhuapl.edu.

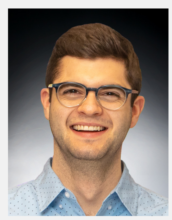
**Miller L. Wilt,** Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Miller L. Wilt is a software engineer in APL's Research and Exploratory Development Department. He holds a BS in computer engineering from Lehigh University and an MS in computer science from Johns Hopkins University. Miller has a diverse set of programming and DevOps skills, including machine learning, full-stack web development, Android development, and software deployment. Recent efforts include investigating the use of machine learning to classify the type of data transmitted over a wireless communications link; leading an assessment of the architecture, design, code quality, and security of a government-developed Android application; and architecting, designing, and developing various applications and microservices. He was awarded three internal IP disclosures resulting in multi-million-dollar follow-on contracts and has received three APL special achievement awards for outstanding technical excellence. His email address is miller.wilt@jhuapl.edu.



**Brock A. Wester,** Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Brock A. Wester is a project manager and supervisor of the Neuroscience Group in APL's Research and Exploratory Development Department. He holds a BS in computer engineering from the Georgia Institute of Technology and a PhD in biomedical engineering from Emory University. Brock's current research efforts span connectomics, neuro big data, neuroprosthetics, user interfaces, and immersion (VR/AR/MR). He is the project manager on the Machine Intelligence from Cortical Networks (MICrONS) project, principal investigator on an NIH grant focused on the development of neuro big-data resources, principal investigator of the Revolutionizing Prosthetics program, and vice chair of the Johns Hopkins Engineering for Professionals Applied Biomedical Engineering program. Before joining APL, Brock was the co-founder and chief technology officer of NanoGrip Technologies, which developed and commercially produced multiple-degree-of-freedom micromanipulation tools for micro-assembly, mechanical testing, and biological interfacing. His email address is brock.wester@jhuapl.edu.



**Jordan K. Matelsky,** Research and Exploratory Development Department, Johns Hopkins University Applied Physics Laboratory, Laurel, MD

Jordan K. Matelsky is a project manager and big-data scientist in APL's Research and Exploratory Development Department. He holds a BS in computer science-neuroscience from Johns Hopkins University. Jordan's current research focus areas include big-data neuroscience connectomics and machine learning for health care environments. His projects include large-scale network analyses on brain maps and central-nervous-system microbiome detection, and he currently serves as a precision-medicine data scientist in collaboration with the Johns Hopkins Hospital system. Prior to joining APL, Jordan was the chief technology officer and cofounder of FitMango, a personalized fitness software company. His email address is jordan.matelsky@jhuapl.edu.