Start Solving with xGraph

November 2018

Typical engineered computer ecosystems are a jungle of applications, raw streaming data, and other systems that must collaborate. Systems you wish worked together include IoT devices, databases, AI, your legacy systems, and more.

xGraph a Technology Solution so you can Stop Coping and Start Solving

Tackling the Complexity of Real-World Problems

Despite decades of bold computing ideas aimed at tackling real-world problems, there remain plenty of unsolved challenges. For every problem solved by relational databases, the standards-based Internet, and object-oriented programming, exponentially more problems are generated every day by big data, the Internet of Things, large-scale AI and the incompatibilities introduced from those solutions. “Wicked problems” such as how to model climate change or how to manage the modern distributed electrical grid can’t be solved with existing monolithic tools. What makes them so tough to solve?

First, these problems entail “systems of systems” that interact in complex ways. These can be much like natural ecosystems with many individual systems working together. This leads to complex problems that often don’t lend themselves to easy solutions with current tools: for several reasons; they need to scale, include unique systems or data structures, need to adapt and change in time, deal with large distributed data sets, and often result in nested systems within systems within systems. Humans, like nature, solve these problems by breaking them into smaller, more specific components.

xGraph is a new category of software that changes the way we approach and solve such complex problems. xGraph is an executable graph – a loosely coupled network of systems that are capable of solving complex, real-time, highly dynamic systems-of-systems problems.

xGraph helps YOU solve these types of problems!

What makes these problems hard to solve?

After years of solving complex, data analytics problems, we have identified the following characteristics that make complex problems so difficult to tackle:

Massive decision spaces. Chess players navigate a decision space of about five million options over the course of an average game; StarCraft has roughly 10^{300}. Today's wicked problems mirror that explosion in decision space. In addition, any one choice can kick off a chain of unintended consequences, meaning complex problems can morph and grow even more complex over time.

High volume/velocity streaming data. Our digital devices are flooding data centers with about two zettabytes of data per year. Much of it is raw data streaming from Internet of Things sensors, and require quick action before the data loses value.

Highly distributed computing power/data collection. We walk around with the equivalent of yesterday's supercomputers in our pockets, yet all of our data flows into the all-too-hackable centralized systems. Very big, interconnected problems can be broken down into multiple smaller (interconnected) problems, with computation done locally (at the edges) rather than in the central facility.

Low latency. When decisions must be made rapidly, local decision making outperforms centralized solutions. Only relevant information is then shared with the broader network mitigating many bandwidth issues.

Dynamic reconfiguration. Accelerating change and adaptability stresses legacy systems that were not designed for dynamic, automated reconfiguration.

Always-on operations. Changes and improvements must happen with no downtime for critical, always-on systems.

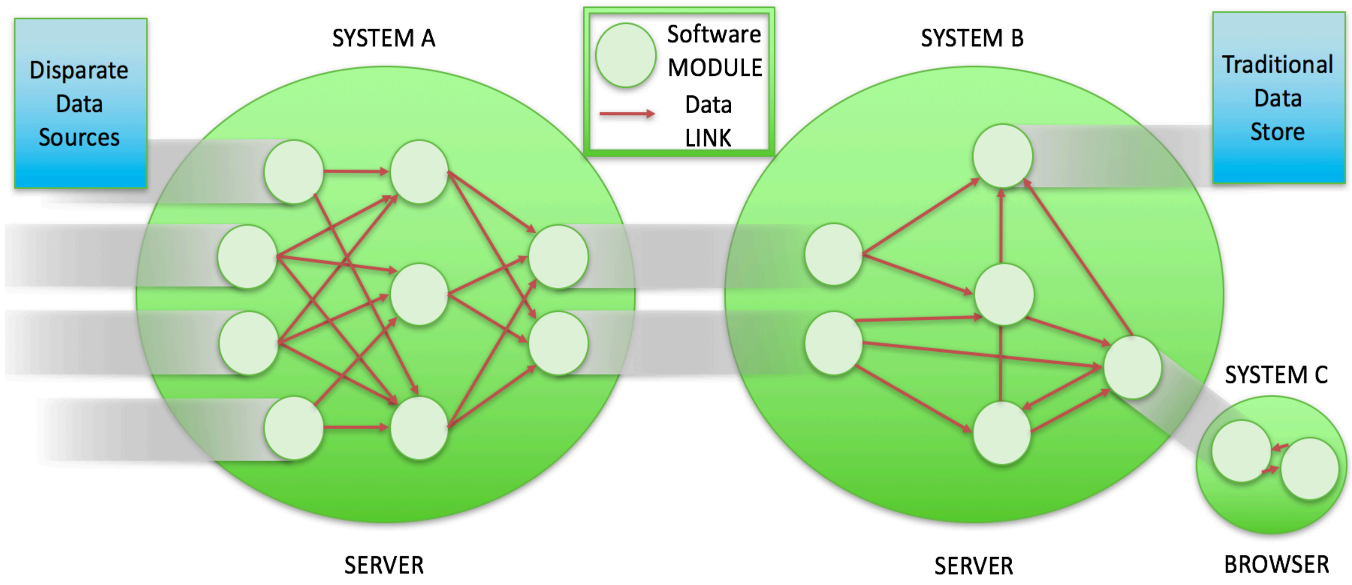
Computational diversity of data and analytics. All data is not equal or even alike, yet many of our biggest challenges rely on a disparate mix that requires analytical diversity and collaborative problem solving.

Collaborative intelligence. Humans interact with machines and autonomous systems on a daily basis. These interactions are not optimized for today's environments or uses.

Many real-world problems share these characteristics. Think about designing "smart cities" able to integrate self-driving vehicles with existing traffic-management systems, electrical grid operators that need to transmit power effectively and limit outages, and managing drone swarms used for package delivery or undersea exploration. Society knows what the problems are – we just need the tools to help us solve them.

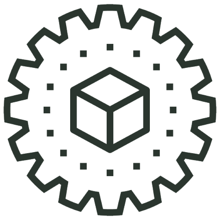
How xGraph is Different

and Why It Can Tackle Wicked Problems

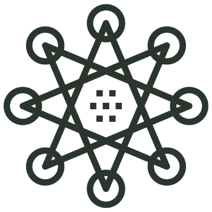


With a patented open-source executable graph platform and a state-of-the-art system-of-systems design environment, xGraph offers a new approach to solution development. xGraph's flexible architecture cleans up messy back-end issues that arise when creating a large-scale, distributed architectures aimed at solving real-time, highly dynamic, systems-of-systems problems. In essence, xGraph solves complex challenges by replicating their structure and data flows. Here's how:

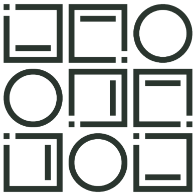
xGraph solutions are composed of nested elements: Entities, Modules, and Systems and Systems of Systems.



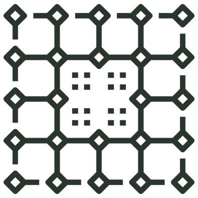
An **Entity** is the most atomic element in an xGraph solution. While hidden from most end-users, the Entities serve as the functional backbone of all deployed applications.



A **Module** is an encapsulated graph of Entities collaborating to perform a service. Modules are the most granular element available to the user when building an xGraph solution.



A **System** is a graph of Modules which constitutes the smallest deployable xGraph application. Systems link Modules (services) to create solutions.



A **System-of-Systems** is a graph of Systems that collaborate in a domain to enable solution architectures in dynamic, complex context.

Entities

Entities can communicate with each other and are defined by the data and functions they contain. They can autonomously modify, connect with, or produce new elements within the xGraph ecosystem as needed.

Event Driven – Event Driven – “Only On when Needed”

Modules

One of the most common services provided by a Module is proxying, which is how data can flow between multiple xGraph Systems that comprise an xGraph solution. In concept, module proxying enables communication to a Module in a remote System (running on a different process) as if it were in the local System. Since xGraph Systems can run in a web-browser environment as well as the server, proxying enables full-stack communication of Modules existing on either side of the server/browser barrier.

Another common service provided by many modules is the wrapping of traditional tools. These include standard software utilities, npm packages, or more robust enterprise software solutions.

Systems

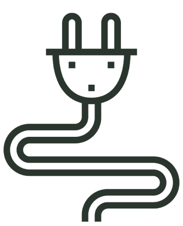
The flexible architectures enabled by xGraph can perform complex calculations at the lowest level, at any number of diverse locations, thus reducing the time to solution. When calculations occur at the local level, security risks are reduced – no more need to send data back and forth via the Internet. Each level can enable autonomous actions based on programmed instructions, and each can interact as needed with each other. When interconnected Systems are multiplied by the thousands, tens of thousands, and more, this environment delivers rapid iteration expediting the time to solution. xGraph’s internal communication protocols are built on JSON, a lightweight data-interchange format. JSON is extremely flexible, and xGraph leverages that flexibility in many of its data models.

xGraph at Work

Today, xGraph is or can be used to solve complex challenges in a variety of real-world applications, including the energy grid, traffic management for smart cities, managing drone swarms for undersea exploration, managing real-time learning across providers in health care, and real-time tracking of seismic events

xGraph has unlimited potential to solve our existing and emerging problems. With our ability to take care of back-end complexity, we make it easy for problem-solvers to create solutions for complex systems of systems.

Stephen Hawking says that the twenty-first century will be the “century of complexity.” We’re ready to solve some wicked complex problems with you.



The Electrical Grid

An electric grid is example of a system-of-systems challenge with a distributed network of micro-services. An xGraph energy hub might manage 20 different electricity-consuming devices within one System. That hub in turn collaborates with a neighborhood management System, which in turn collaborates with a city-wide management System, and so on up the scale. The lower devices can make autonomous decisions rather than relying on upper systems to control the interactions.

This type of collaborative AI streamlines decision-making, slashes processing time, and effectively communicates information, enabling all Systems to function efficiently. Most of the world's most complex problems manifest within these systems of systems; they are fractal in nature. xGraph uniquely addresses complex problems with this approach.

This fractal application structure has several benefits:

- **Reliability** – Systems are independent and promote self-healing characteristics, making the overall system resilient.
- **Security** – What happens in a smaller System stays in the smaller System. Just like Vegas.
- **Heterogeneity** – Even if the global goal is similar – reliable electricity at lowest cost – each System can adapt at its own scale.
- **Appropriate interdependence** – When connections must occur, they do so at the proper level of metadata rather than trying to optimize centrally based floods of data.

xGraph has unlimited potential to solve our existing and emerging 'wicked problems'. With our ability to take care of back-end complexity, we make it easy for problem-solvers to create solutions for complex systems of systems.