






Company

Founded in 2003 Zethus Software is focused on developing software products using grid computing technologies to solve real-world business problems.

Mission: Design and develop software tools that allow users to concentrate on their work, not operate computer systems. There is a sweet spot in software development where innovation and theoretical work can be applied to real-world problems to produce an amazing, positive impact on a person's ability to perform a task – the system becomes transparent and the task illuminated.

Zethus Software has created a solid set of technologies and tools that allow us to design and create distributed yet highly connected applications with surety.

Product History

	2004-2006	Grid computation and work flow environment
	2005	Web infrastructure load and regression testing tool – Grid technology.
	2006	Web based sales and marketing tool. Interactive
	2006-2007	Client server job quotation system.
	2007	Massive virtual data archive composed of loosely organized devices – cloud computing. Virtualized, de-duplicated, shredded and scattered data storage.

Technology




Zethus Software's technology has evolved since founding but the intent has remained focused: Allow a loose collection of computing devices to work together to provide unified, cohesive services with zero or minimal administration. Terms like distributed, grid, pervasive or cloud have been used over the past decades to describe this type of computing system. We embraced the term grid computing at our inception but lately have been favoring cloud computing. Grid computing speaks to interchangeable, transient resources and services that can be plugged into the grid – which we still embrace. Cloud computing speaks to the loose, amorphous organization – which we seek in our current generation core technology.

The technology (concepts and core principles) and how it is implemented (expressed in machine executable code) are complimentary not inextricably tied. Our current generation of technology is implemented in ObjectPascal, Java, JavaScript, Python and PHP. We are transitioning to more Java and some C with JavaScript, Python and PHP used in special cases.

We currently have over 300K lines of code that represent our frameworks and products. This is a metric of limited value since it does not represent the true value of the technology which is how the concepts and innovations are designed. The lines of code only represent the current expression of the design. The number of lines of code does give a rough measure of the complexity and scale of the effort to codify the concepts.

The primary value that Zethus Software brings to the project is our experience translating real-world problems into robust, flexible software system designs that can be expressed in code to produce practical, working solutions. Starting with an existing well designed framework and design principles accelerates and frames the solution.

Technology History

	Foundation Framework	Establishes base functionality to build higher order elements
	Generation One Hub organized grid	Designed and implemented as grid workflow system.
	Generation Two Cloud computing	Quorum and Archive constitute base services.

Development Principles

- ♦ Highly distributed, loosely organized – this allows nodes to come and go, be added or removed without adversely affecting the system as a whole. This also adds to the robustness of the system – no single points of failure or attack.
- ♦ Service Oriented Architecture (SOA; machine-to-machine automation) – highly interfaced for extensible, collaborative development and maintenance. Well defined interfaces between modules and elements – coding becomes less important and more flexible and sustainable. User interfaces just become additional service calls to the system, but now systems can be automated since all functions are service calls that can be programmed.
- ♦ Modular, decoupled functionality – functions are separated into distinct modules with well defined interfaces which allows better design, re-factoring, and extensibility.
- ♦ Scalable – grows without systemic alteration
- ♦ Extensible – altered without systemic alteration
- ♦ Pipeline – flexible, content agnostic. This can also be described as data-centric design. The system acts more as a pipeline – moving data through the system having specific understanding of the data. Data is used to define data (meta-data);
- ♦ Web launched Rich Internet Application (RIA) – as available as common browser based interfaces, but richer in capabilities since they are not constrained by the browser’s capabilities.
- ♦ Use open source code when possible – leverage existing code and frameworks where it makes sense and does not create licensing issues.
- ♦ Cross platform code foundation (e.g. Java, Python, etc.)
- ♦ Semantic messaging (e.g. XML) – This is the implementing method for the system’s distributed, loosely coupled SOA. State and commands are communicated via messages that have semantic data.

Pre-existing Intellectual Property, Concepts and Code

Zethus Software will license its pre-existing intellectual property (IP), concepts and code to accelerate and frame the CyberLabNet project. This will constitute a license to use the IP, concepts and code with the CyberLabNet project, tools and services. Zethus Software will design and implement software systems in collaboration with YSU for the CyberLabNet project. This shared IP and code will be of shared ownership and fall under the terms of the subcontract to be negotiated. Zethus Software will keep a bright line separation between IP and code that it owns and that which is shared.

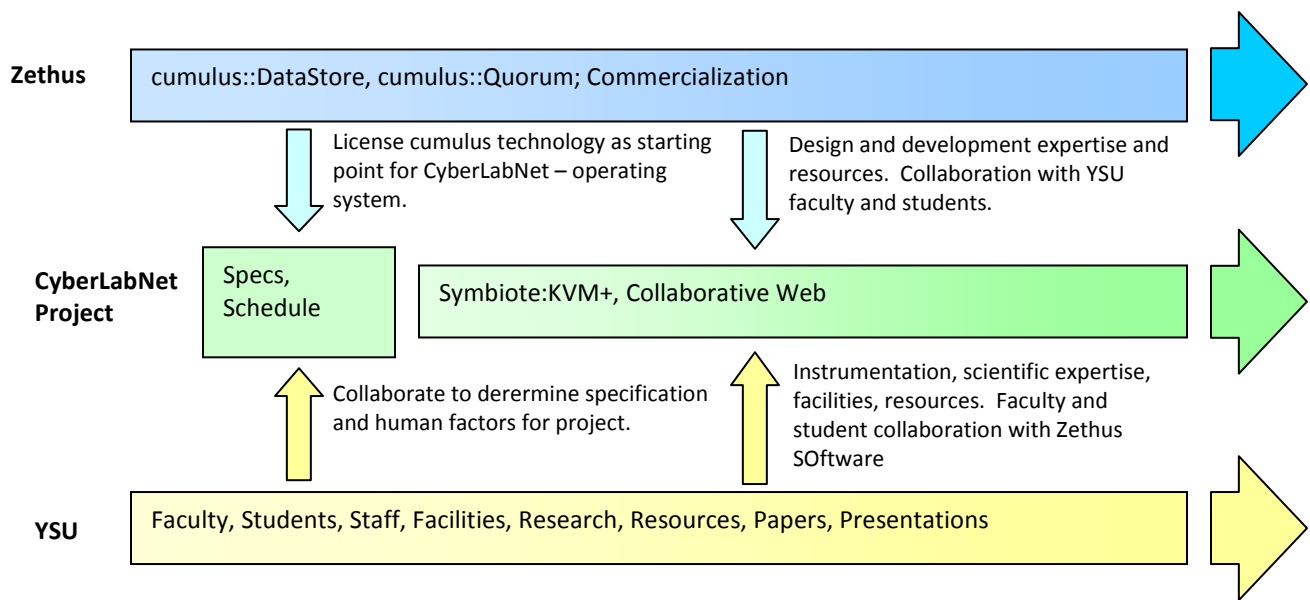


Diagram showing the contributions, collaboration and owned and shared IP by organization. This acknowledges that each organization (Zethus and YSU) will continue develop and retain ownership of existing IP while also collaborating to develop new IP that will benefit YSU and Zethus. Zethus is ready to work with YSU to enter into a contract stipulating the details of ownership of this new IP developed during the collaboration.