Memorandum on Canvas* Research

Block Science
January 17th 2025

Orion Reed
Senior Researcher
orion@block.science

Jeff Emmett
Project Manager
jeff@block.science

*tentative proposal to rename this project to 'interlay' from *inter-* (among, between) and *layer*.

Introduction

Today's software follows an application-centric model in which applications act as silos of data and behavior. This model is produced and maintained by economic dynamics that favor software producers over end-users, creating an artificial barrier to participation in the design and organization of digital information.

This research project aims to intervene in how *software interfaces* are constructed as a means of changing the 'topology' of the app model, decoupling interfaces from applications and creating a new *interface integration domain* to enable the self-organization of information and behavior across existing software boundaries. This work was born out of the unique genericism of the KOI and RID research at BlockScience, and will leverage both the tech and theory of that work.

Problem Statement

The application model, in which software is analogous to *commercial appliances*, is the result of a software-as-commodity economic model [1]. The siloing of information, lack of interoperability, opacity to inspection and inflexibility for end-user modification is a downstream product of this model - these are features, not bugs.

The app model poses significant challenges for knowledge workers whose economic value is intertwined with data and skills related to specific applications and whose access to this value is systematically jeopardized in collaboration due to the different application practices, preferences, and proficiencies of other stakeholders [2]. It also fails to support the "long tail of user needs" as it can only target the biggest problems for *most* people, leaving the majority of needs out of scope [3].

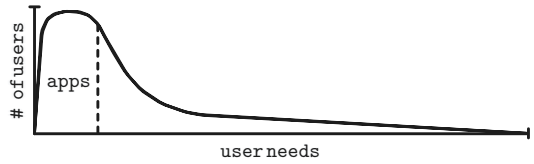

Figure 1: *The long tail of user needs*

As such, several movements have grown up around these failings: *local-first software* aims to "bring the services to the data" in contrast to today's cloud services, where data lives within products in remote server farms [4]; *malleable software* aims to create software which is "as easy to change, as it is to use" [5]; *interop advocacy* aims to push for changes in law supporting open and interoperable data [6].

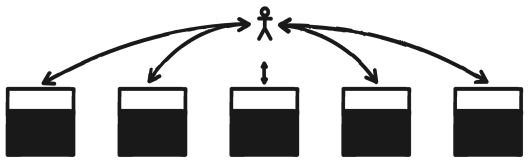

Figure 2: *Apps as appliance-like information silos*

This research is based on a belief that the problem is not technical but eco-

nomic - that the process of producing software, the *mode of software production* [7], is the root cause of the app model's persistence. As such, the goals of our technical interventions are aimed to shift this economic arrangement by reducing end-user reliance on commercial software providers.

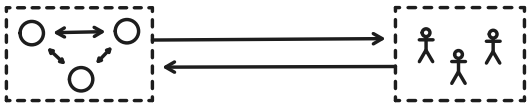

Figure 3: *Mode of software production as defined by a combination of relations of production - the way people organize themselves around labor - and forces of production - the means of labor, such as tools and techniques.*

Prior Work

One promising approach to understanding the rigidity of apps is the lack of an *integration domain* for software, which is simply a set of languages or tools for performing integration. Informally, many ad-hoc special-purpose integration domains have emerged in conventional practice - for example, scripting languages (like the Unix shell) or patching tools (like Unix's diff). Each of these is highly specialized and highly constrained, but hint at the more general and powerful tools [8]. We see an under-exploited opportunity to apply this work to software interfaces.

The intertwined concepts of *computational media* [9] and *information substrates* [10] provide both useful metaphors (how we want software to *feel*) and a theoretical grounding. The "computation as a medium" metaphor dates back to the 1970s and draws from physical media such as paper and clay, whereas substrates draw inspiration from biology, as the material from which an organism lives and grows - like the soil in which mycelial networks form.

Computational media seeks to create software that is "malleable, shareable, distributable, and computable." *Information substrates* build on this, highlighting that "data does not exist in a vacuum; it is part of a substrate that provides context for its interpretation and constraints for presenting and interacting with it." Examples of substrates include musical scores, spreadsheets, or layers in tools like Photoshop. By reorienting around *computational materials* and separating the *tools* and *instruments* from the *media* they act on, we can get far more heterogenous and flexible interfaces [11]. This reorientation may even be a prerequisite for achieving true computational literacy [12].

Research Thesis

Current *software interfaces* create a software topology which is resistant to change by reinforcing a 1-1 relationship between *system* and *user interface* [13] and requiring steep coordination among interface authors which incentivizes designs that are not amenable for reuse and recomposition [14].

The thesis of this research is that we can create an *integration domain for interfaces and interaction* which will expand the use-value of existing software while lessening the forces which reproduce the application model, giving end-users more agency over their computational environments.

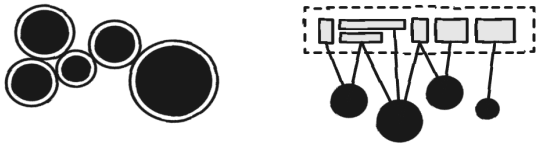

Figure 4: *App-centric (left) vs the 'networked' (right) software topologies.*

Specifically, we are exploring **spatial canvases** (a set of shapes with positions in an infinite Euclidean plane) and **extensions to native web stan-**

dards such as **Web Components** and the **Geometry Interfaces API** as specific technical approaches. Building on web standards reduces the entry barrier to simple .html files with all required technologies already included in a user's browser. Deep addressability (via CSS selectors) is of principal importance to malleability [14] and the basis for heterogenous organization [15], one of the core motivators behind RIDs. Lastly, spatial canvases enable *geometric encapsulation* (behavior encapsulated to the interior of 2D geometry) which reduces coordination requirements among authors.

Figure 5: *Tech stack comprised of largely independent primitives built atop native web standards.*

Where the web cannot meet our needs today, we extend the client-only tech stack with other software such as semi-secure cross-origin key handling via a **secureFetch** browser extension, open network peer discovery using BitTorrent and self-hosted BlockScience servers.

We are working to create *inter-systemic primitives* which exist among or between existing systems, analogous to connective tissue, and which can be used independently of each other.

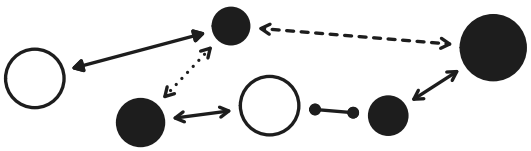

Figure 6: *Inter-systemic primitives.*

Results to date

Semantic Morphisms: structure-preserving interpolation between different views or representations.

Scoped Propagators: a novel programming model based on event-driven state delta propagation along graph edges, amenable for embedding into existing interfaces and shown to be generally quite terse and expressive. This work has been adapted by several others and was accepted to ACM SPLASH in 2024 [16]

Social Expression Graphs: a spreadsheet-like data model allowing the input and aggregation from multiple actors which can be processed to create governance mechanisms and richer social inputs to software.

Effect Integrators: a generic approach to combine multiple running simulations acting on the same set of objects without coupling.

Other results include *localized layout constraints*, the *BYOA (bring-your-own-access) model* of security and API access, *WeakLayout* primitives and *BitTorrent peer discovery*, among others.

Conclusion

This memo is a notice of intent to do research and *make some cool shit*.

Figure 7: *Research results are bundled up into projects or products when sufficient product-market-fit is realized for specific capabilities*

We aim to produce communicable and publishable research while using research outputs to produce versioned and stable (but not necessarily future-compatible) tools for BlockScience in sandboxed environments. Results of this research can be increasingly focused toward external, revenue-generating projects and grant-funded projects as opportunities permit.

Bibliography

[1] E. Elkins and J. Morris, "There's a History for That: Apps and Mundane Software as Commodity.," *The Fibreculture Journal*, Jan. 2015, Accessed: Oct. 05, 2023. [Online]. Available: https://www.academia.edu/19579663/There_s_a_History_for_That_Apps_and_Mundane_Software_as_Commodity

[2] M. Nouwens and C. N. Klokmose, "The Application and Its Consequences for Non-Standard Knowledge Work," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC Canada: ACM, Apr. 2018, pp. 1–12. doi: 10.1145/3173574.3173973.

[3] M. Appleton, "Home-Cooked Software and Barefoot Developers." Accessed: Jan. 16, 2025. [Online]. Available: https://maggieappleton.com/home-cooked-software/

[4] M. Kleppmann, A. Wiggins, P. Van Hardenberg, and M. McGranaghan, "Local-First Software: You Own Your Data, in Spite of the Cloud," in *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Athens Greece: ACM, Oct. 2019, pp. 154–178. doi: 10.1145/3359591.3359737.

[5] "Malleable Systems Collective." Accessed: Jan. 16, 2025. [Online]. Available: https://malleable.systems/

[6] C. Doctorow, "Adversarial Interoperability." Accessed: Aug. 09, 2020. [Online]. Available: https://www.eff.org/deeplinks/2019/10/adversarial-interoperability

[7] P. Tchernavskij, "Designing and Programming Malleable Software," 2019.

[8] S. Kell, "The Mythical Matched Modules: Overcoming the Tyranny of Inflexible Software Construction," in *Proceeding of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications - OOPSLA '09*, Orlando, Florida, USA: ACM Press, 2009, p. 881. doi: 10.1145/1639950.1640051.

[9] M. Borowski, "Exploring Computational Media as a Possible Future of Software," 2022.

[10] M. Beaudouin-Lafon, "Information Substrates."

[11] M. Beaudouin-Lafon, "Beyond Applications: Interaction Substrates and Instruments," in *Proceedings of the 34th Conference on l'Interaction Humain-Machine*, in IHM '23. New York, NY, USA: Association for Computing Machinery, May 2023, pp. 1–15. doi: 10.1145/3583961.3583968.

[12] A. diSessa, *Changing Minds: Computers, Learning, and Literacy*. The MIT Press, 2000. doi: 10.7551/mitpress/1786.001.0001.

[13] O. Reed, "Towards an 'Integration Domain' for HCI." [Online]. Available: https://www.orionreed.com/artifact/tft-rocks-integration-domain.pdf

[14] A. Basman, C. Lewis, and C. Clark, "The Open Authorial Principle: Supporting Networks of Authors in Creating Externalisable Designs," in *Proceedings of the 2018 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, in Onward! 2018. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 29–43. doi: 10.1145/3276954.3276963.

[15] O. Reed, "Objects as Reference."

[16] O. Reed, "Scoped Propagators." Accessed: Jan. 16, 2025. [Online]. Available: https://www.orionreed.com/posts/scoped-propagators