

Toward Securely Programming the Internet

Andrew C. Myers

Department of Computer Science, Cornell University, Ithaca NY 14853, USA

andru@cs.cornell.edu

<http://www.cs.cornell.edu/andru>

Abstract. Computation and persistent storage are rapidly moving into the distributed domain. Yet we are offered very weak security and privacy assurance, especially as complex information systems share information across trust boundaries. A fundamental problem is that these systems are constructed at too low a level of abstraction. Higher-level abstractions are needed for building complex distributed information systems securely and compositably.

Fabric [1] is a new decentralized platform that embodies this approach. Heterogeneous, mutually distrusting network nodes can securely share both information and computation. Its high-level programming language makes distribution and persistence largely transparent to programmers. However, its Java-like object model is extended to label data resources with explicit confidentiality and integrity policies. Exposing these policies to programmers and to the underlying run-time system enables programmers to reason about security, and enables the system to enforce them through a combination of compile-time and run-time mechanisms. Optimistic, nested transactions ensure consistency across all objects and nodes. A peer-to-peer dissemination layer helps to increase availability and to balance load. Results from applications built using Fabric suggest that Fabric has a clean, concise programming model, offers good performance, and enforces security.

Reference

1. Liu, J., George, M.D., Vikram, K., Qi, X., Wayne, L., Myers, A.C.: Fabric: a platform for secure distributed computation and storage. In: Proc. 22nd ACM Symp. on Operating System Principles (SOSP), pp. 321–334 (October 2009)