Radiant object vector distribution

John D.H. Pritchard [*]

January 19, 2021

Abstract

Radiant object theory [4, 9, 5, 6, 8, 7] opened a perspective on social peer trust networks. Radiant object vectors [10] is an evolution of that examination of distributed objects. This effort is a review of that work.

1 Structure

Social peer topology is typically exemplified by `git` [13]. Object reference targets are (functionally equivalent to) persistent and immutable content (version) hash symbols (like git commit identifiers). Collections are software project directory structures. Distribution is push - pull agnostic.

The radiant object vector is like a collection of `git` projects in that the radiant object is like a software project directory structure [4, 10]. In radiant object theory, distribution is push - pull agnostic. Like `git`, peer topology is distributed. Unlike `git`, radiant object vectors need a fidelity of immediate continuity for the integrity of local order consistency.

Each endpoint is able to determine logical integrity constraints to manage the consumer store when application references are object constrained. When an object is a software project directory structure [2], dependencies are constrained to the object like a self contained HTML or SVG [11, 3]. Self contained objects are endpoint managed [1, 12]. Object vector integrity is regional [10].

The general shape of push - pull agnosticism is that interaction is an endpoint process independent of communication [5].

$$\pi_S^\lambda \leftarrow \sum_{\tau\kappa}^{\rho\nu} \pi_D^\alpha$$

A trusted peer contact may include an interaction, which may yield a distributed broadcast occasion according to store and forward architecture and management.

Unlike `git`, radiant object distribution yields a producer cache store vector that requires local continuity and consistency. As a result, the push - pull distribution event is a synchronization occasion constrained by vector, store, and forward architecture and management. Unlike `git`, rewriting a radiant object produces an independent object.

[*] @syntelos, logicalexistentialism@gmail.com

References

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838, IETF, April 2007.

[2] Ned Freed and Nathaniel S. Borenstein. Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies. RFC 2045, RFC Editor, November 1996.

[3] Dean Jackson and Craig Northway. Scalable vector graphics (SVG) full 1.2 specification. WD no longer in development, W3C, April 2005. https://www.w3.org/TR/2005/WD-SVG12-20050413/.

[4] J. Pritchard. Introducing radiant object theory. Technical Report 20201212, Syntelos, December 2020.

[5] J. Pritchard. Radiant object applications. Technical Report 20201214, Syntelos, December 2020.

[6] J. Pritchard. Radiant object continuity. Technical Report 20201215, Syntelos, December 2020.

[7] J. Pritchard. Radiant object integrity. Technical Report 20201218, Syntelos, December 2020.

[8] J. Pritchard. Radiant object security. Technical Report 20201217, Syntelos, December 2020.

[9] J. Pritchard. Radiant object topology. Technical Report 20201213, Syntelos, December 2020.

[10] J. Pritchard. Introducing radiant object vectors. Technical Report 20210111, Syntelos, January 2021.

[11] Dave Raggett, Ian Jacobs, and Arnaud Le Hors. HTML 4.0 recommendation. W3C recommendation, W3C, April 1998. https://www.w3.org/TR/1998/REC-html40-19980424/.

[12] K. Scott and S. Burleigh. Bundle protocol specification. RFC 5050, RFC Editor, November 2007.

[13] Linus Torvalds. The git software version control system. Technical report, Linux Foundation. https://git.kernel.org/.