COAST: An Architectural Style for Decentralized On-Demand Tailored Services

Michael M. Gorlick, Kyle Strasser and Richard N. Taylor
Institute for Software Research
University of California, Irvine

WICSA/ECSA 2012
Helsinki, Finland
24 August 2012

Context: Decentralized Computation

- Distributed computation among multiple spheres of authority
  - *Disaster response (Hurricane Katrina, New Orleans, August 2005)*
    - National, regional, state, local, NGOs, volunteers
  - *Large-scale engineering*
    - Boeing 787 Dreamliner or Airbus 350 XWB
  - *Scientific computing*
    - Bioinformatics (computational genomics or proteomics)
  - *Weather forecasting*
    - Many sensor networks
    - Many models
  - *Computational health care*
    - Data-intensive personalized medicine *(The Atlantic, July/August 2012)*
  - *Logistics*
  - *Just-in-time manufacturing*

Simultaneous increase in both diversity and integration

Decentralized Computation: Many Paths

- Mastery of data exchange, RPC/RMI, and client-side scripting dominates decentralized applications
  - *MapReduce, Hadoop, Picollo (Power & Li, "Piccolo: Building Fast, Distributed Programs with Partitioned Tables," OSDI, 2010)*
  - *Globus, Condor (Thain, Tannenbaum & Livney, "Distributed Computing in Practice: The Condor Experience," Concurrency: Practice and Experience, 2004)*
  - *CORBA (RPC), Java (RMI), Erlang (message-passing)*
  - *Ajax, Yahoo Pipes, Mashlight (Albinola et. al., "Mashlight: a Lightweight Mashup Framework for Everyone," WWW 2009)*
- Our approach to decentralized computation has evolved
  - *Khare & Taylor, "Extending the REpresentational State Transfer (REST)Architectural Style for Decentralized Systems," ICSE, 2004*
  - *Erenkrantz, Gorlick & Taylor, "From Representations to Computations: the Evolution of Web Architectures," FSE, 2007*
  - *Erenkrantz, "Computational REST: A new model for Decentralized, Internet-Scale Applications," PhD thesis, University of California, Irvine, 2009*

Goals and Means

- Internet-scale decentralized applications
  - *Adaptivity*
  - *Flexibility*
  - *Agility*
  - *Safety*
    - Secure communications and information
    - Protect host computing resources
    - Defined valued organizational assets
      - *Data bases, sensors, algorithms, users*
- Means
  - *Stylistic rules*
  - *Bound behavior of mobile code with architecture-centric mechanisms*
    - Principle of Least Authority (POLA)
    - Capability-based security
  - *Safety through mobile code*

Decentralized Computation: A Different Approach

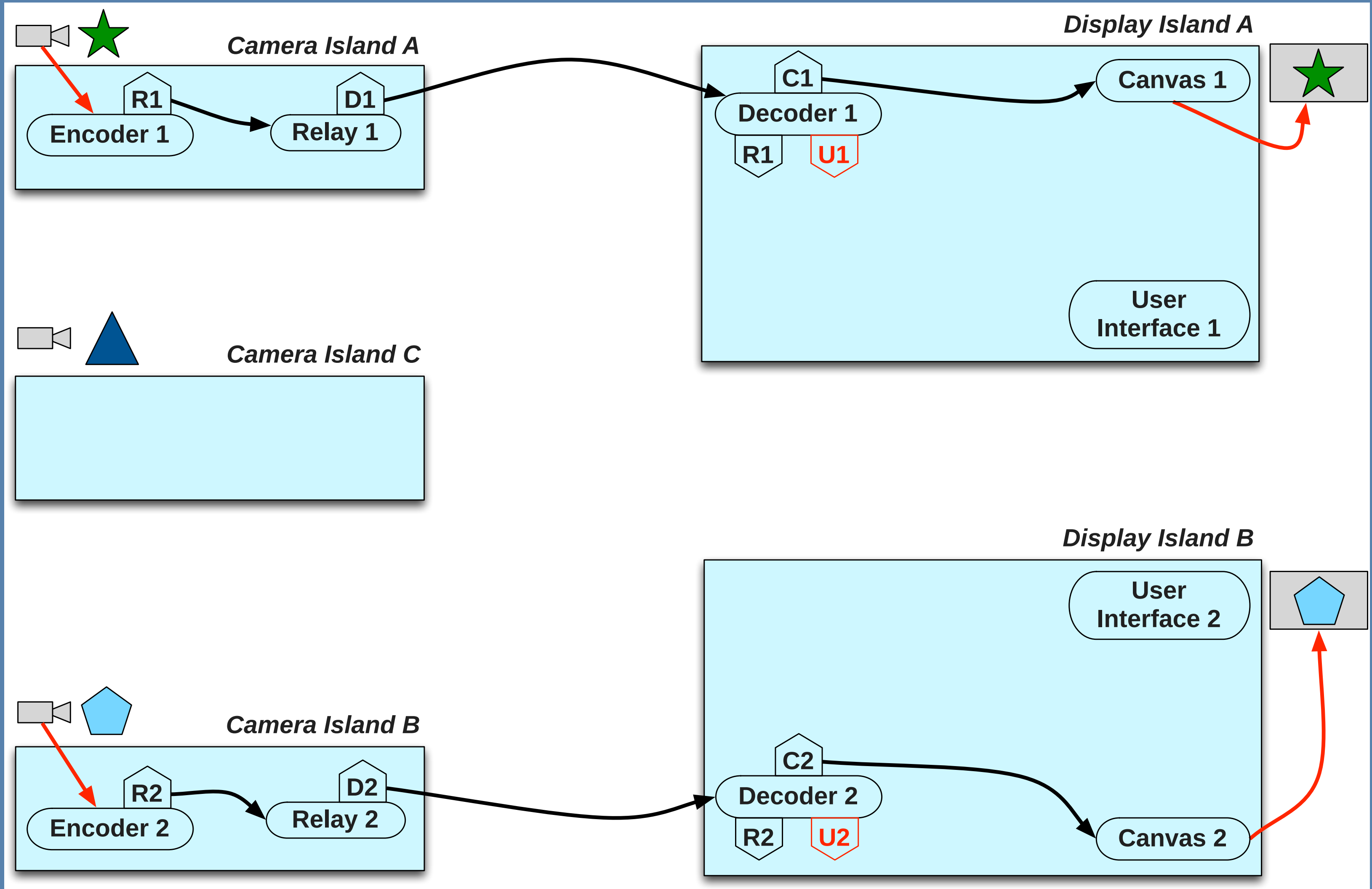
- Exchange **active** computations among peers
  - *Code + run-time state (reified as closures and continuations)*
- Novel security mechanism: Capability URL (**CURL**)
  - *Dictates where computations may go and how they communicate*
  - *Bounds what visiting computations can do*
  - *Limits resource consumption of computations*
  - *Enforces complex constraints*
- Architectural style: COmputAtional State Transfer (**COAST**)
  - *Build **capability** security into the architectural style*
    - Functional capability
      - *What can a visiting computation do?*
    - Communication capability
      - *With whom may that computation communicate?*
      - *When may that computation communicate?*
      - *How often may that computation communicate?*

Architectural style can induce application security

COAST Design Intuitions

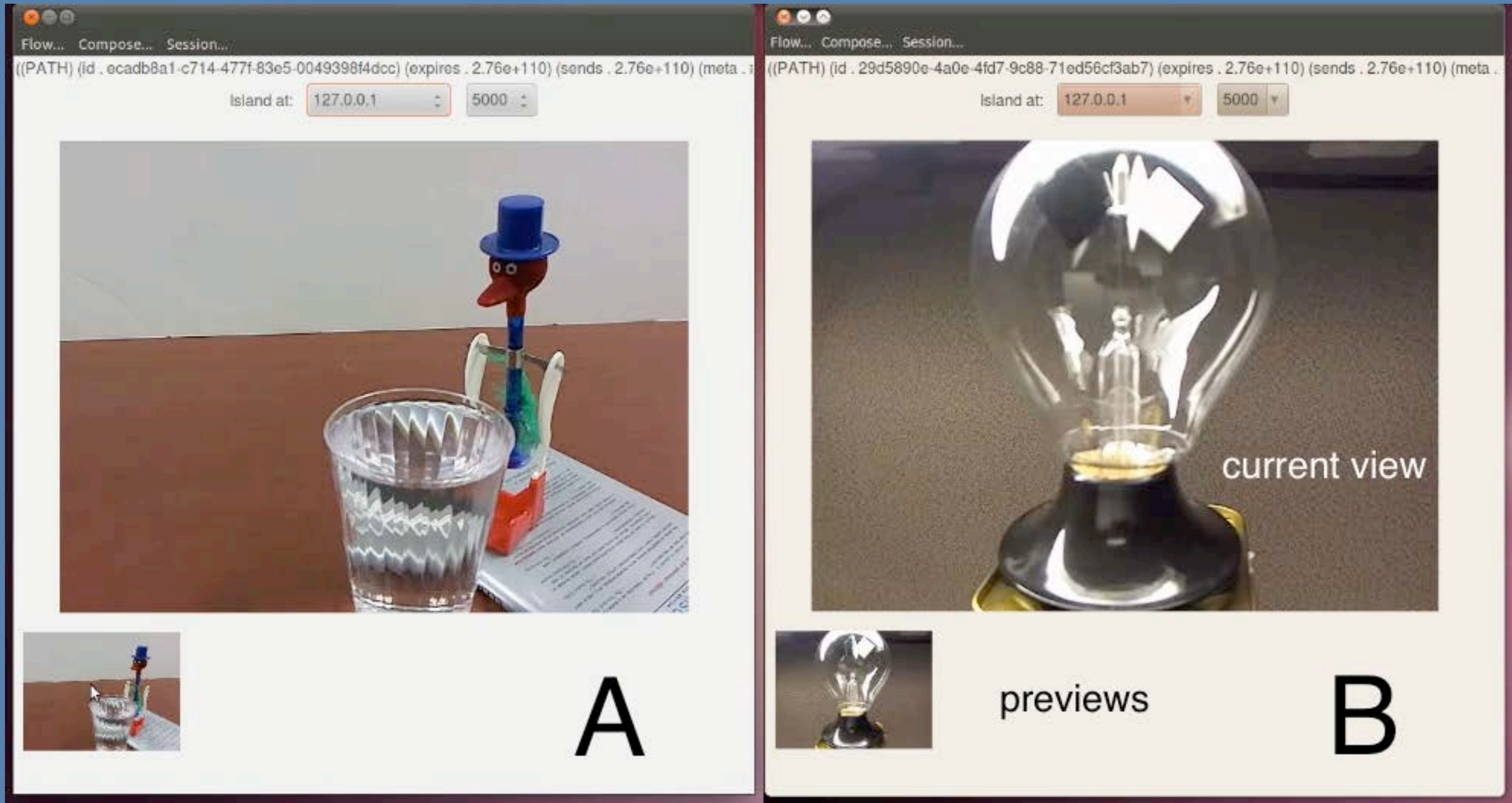
- Computations
  - *Factor your application into many collaborating computations*
  - *Computations are cheap*
  - *Move computations to assets: processors, data, bandwidth, sensors ...*
  - *Computations isolated from one another except by message-passing*
- CURLs
  - *Convey the right to communicate*
  - *Can not be guessed or forged and are tamper-proof*
  - *Carry limitations (time-limited offers, single-use, non-delegable, ...)*
  - *Revocable by issuer at any time*
  - *Critical to the COAST security model*
- Challenge problem
  - *Soft real-time video distribution*
    - Many cameras to many consumers
    - Video sharing and manipulation

COASTcast: A Real-time Video Distribution Application



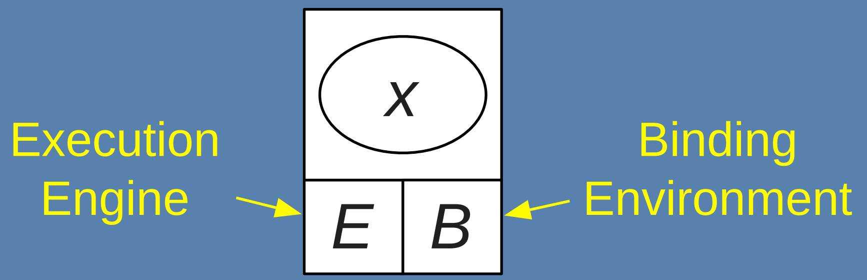
Animation #1: Video from camera to display

COASTcast The Movie: Two Separate Video Flows



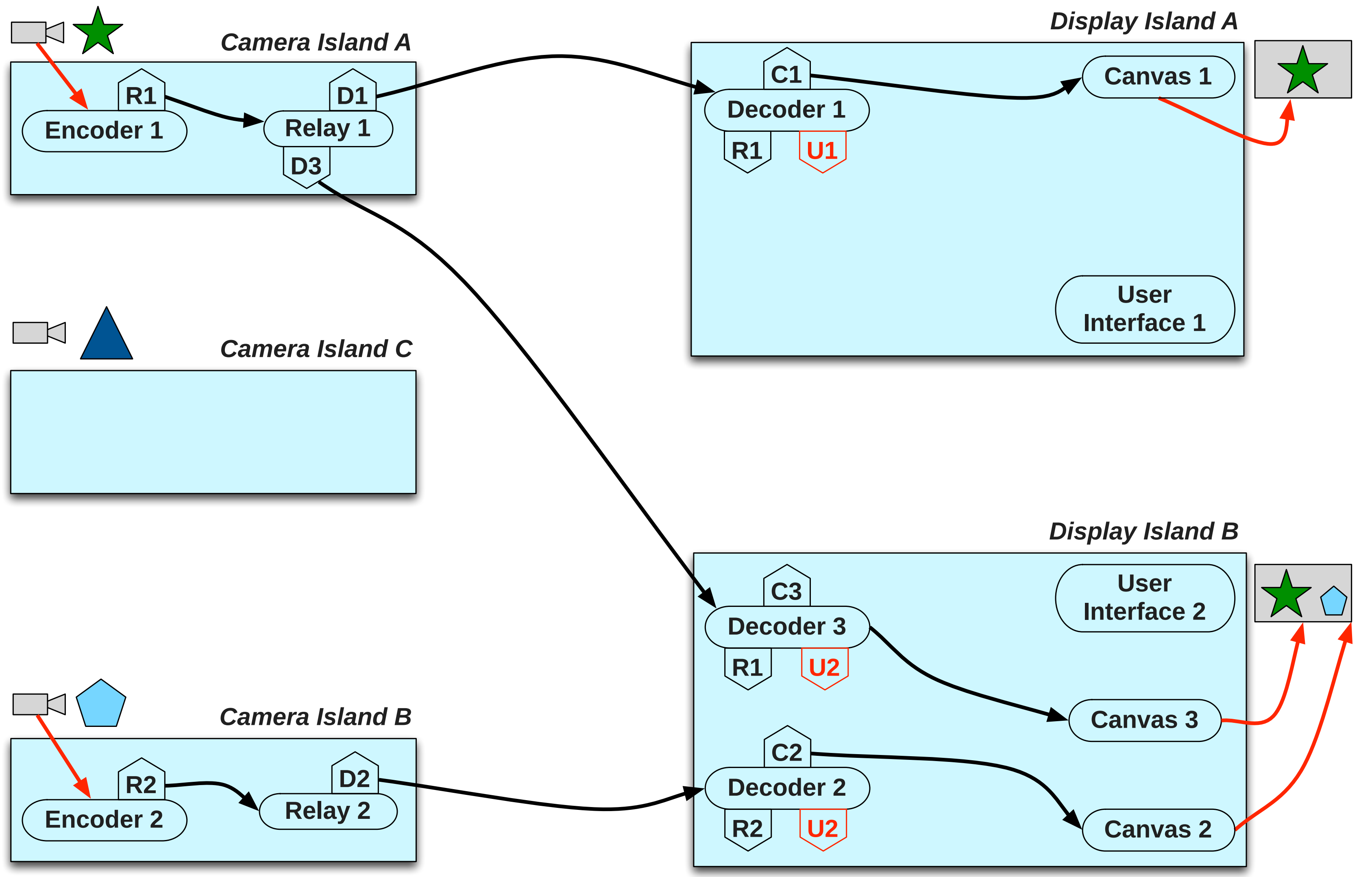
COAST: The Architectural Style

- Applications are comprised of *computations* whose sole means of interaction is the *asynchronous messaging* of *closures*, *continuations*, and *binding environments*
- All computations execute within the confines of some execution site



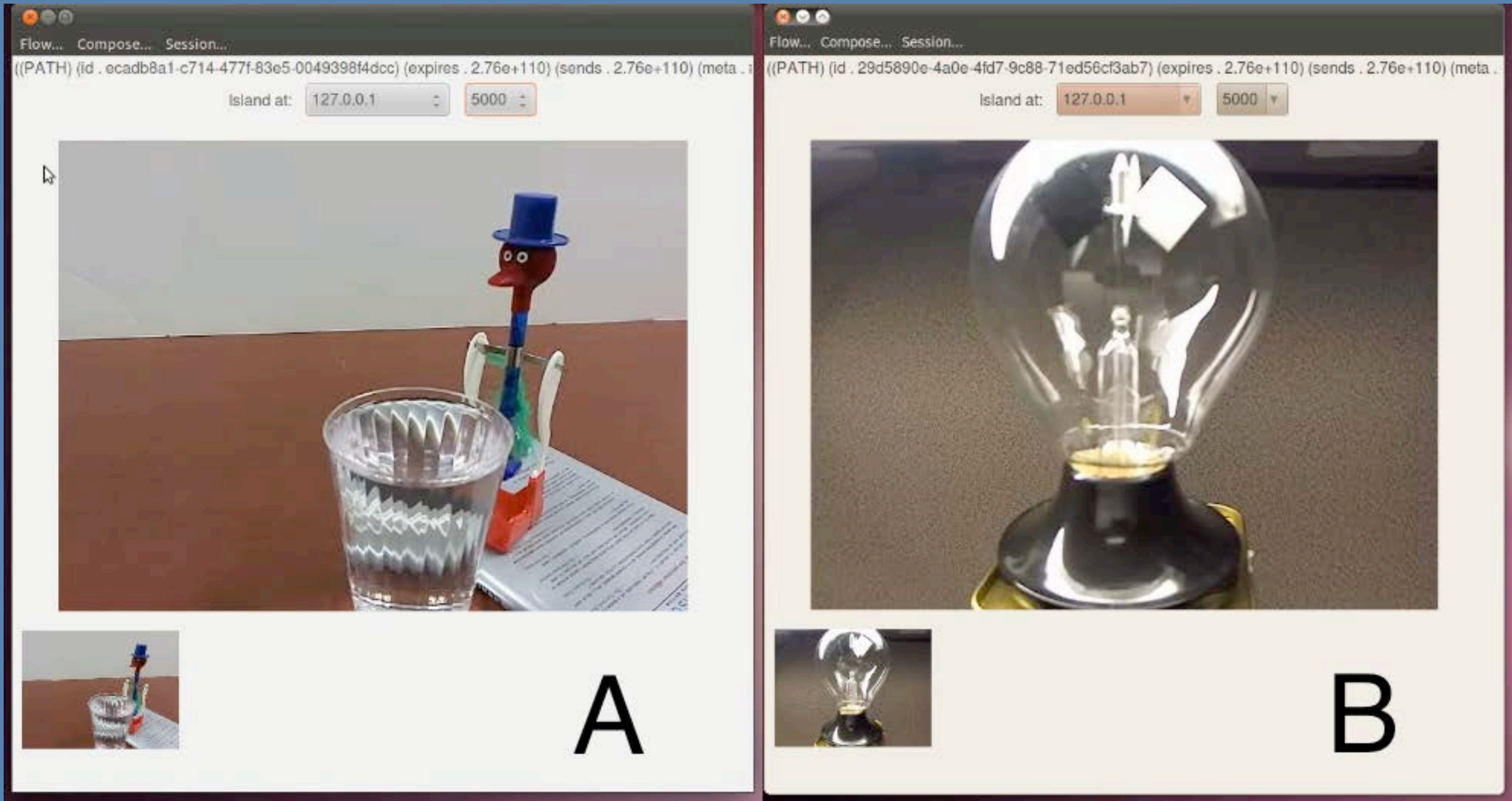
- Computations are named by Capability URLs (CURLs)
  - *Computation **x** may deliver a message to computation **y** only if **x** holds a CURL **u** of **y***
  - *The interpretation of a message **m** delivered to computation **y** via CURL **u** of **y** is **u**-dependent*

COASTcast: A Real-time Video Distribution Application

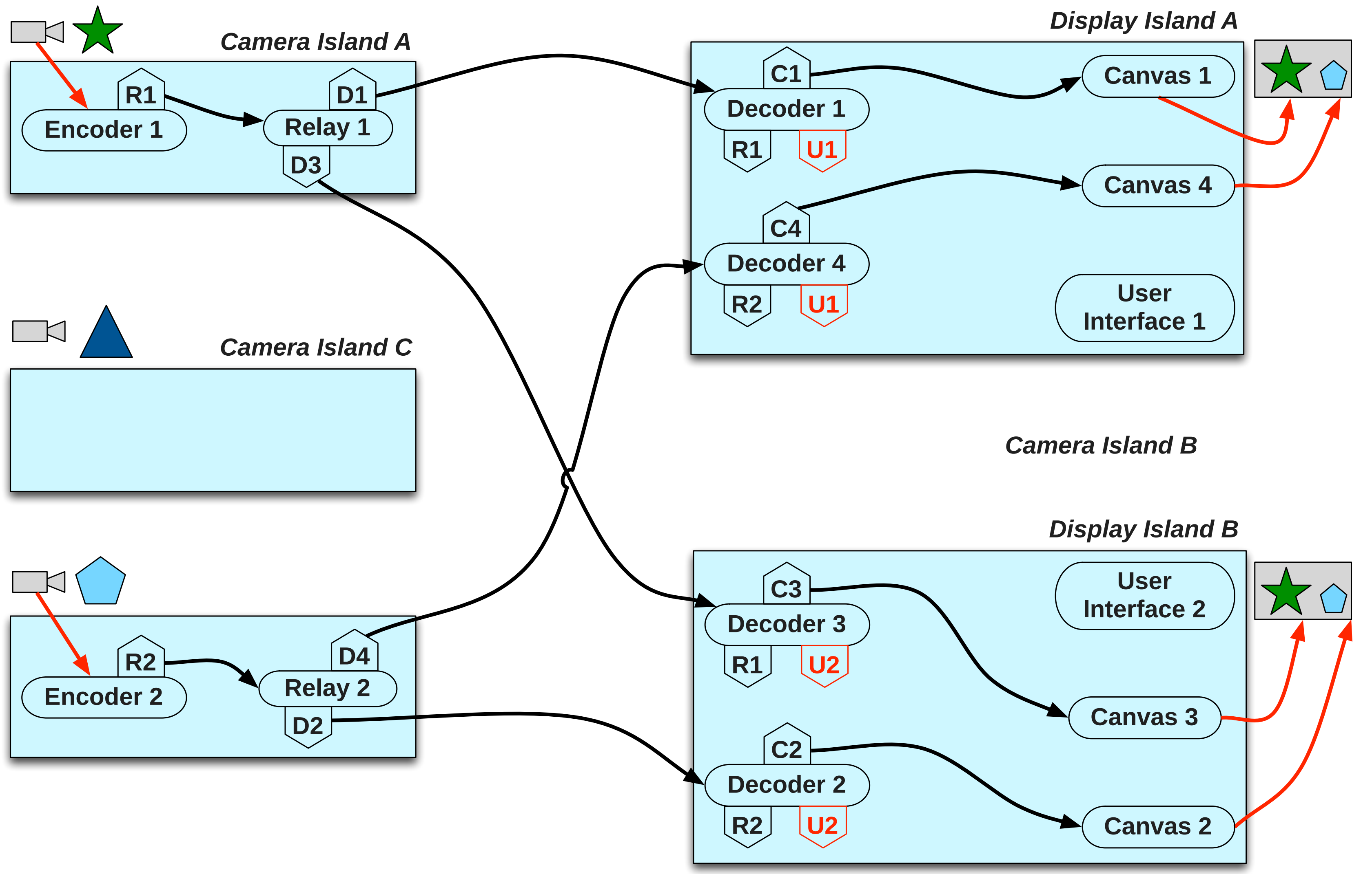


Animation #2: Sharing Video

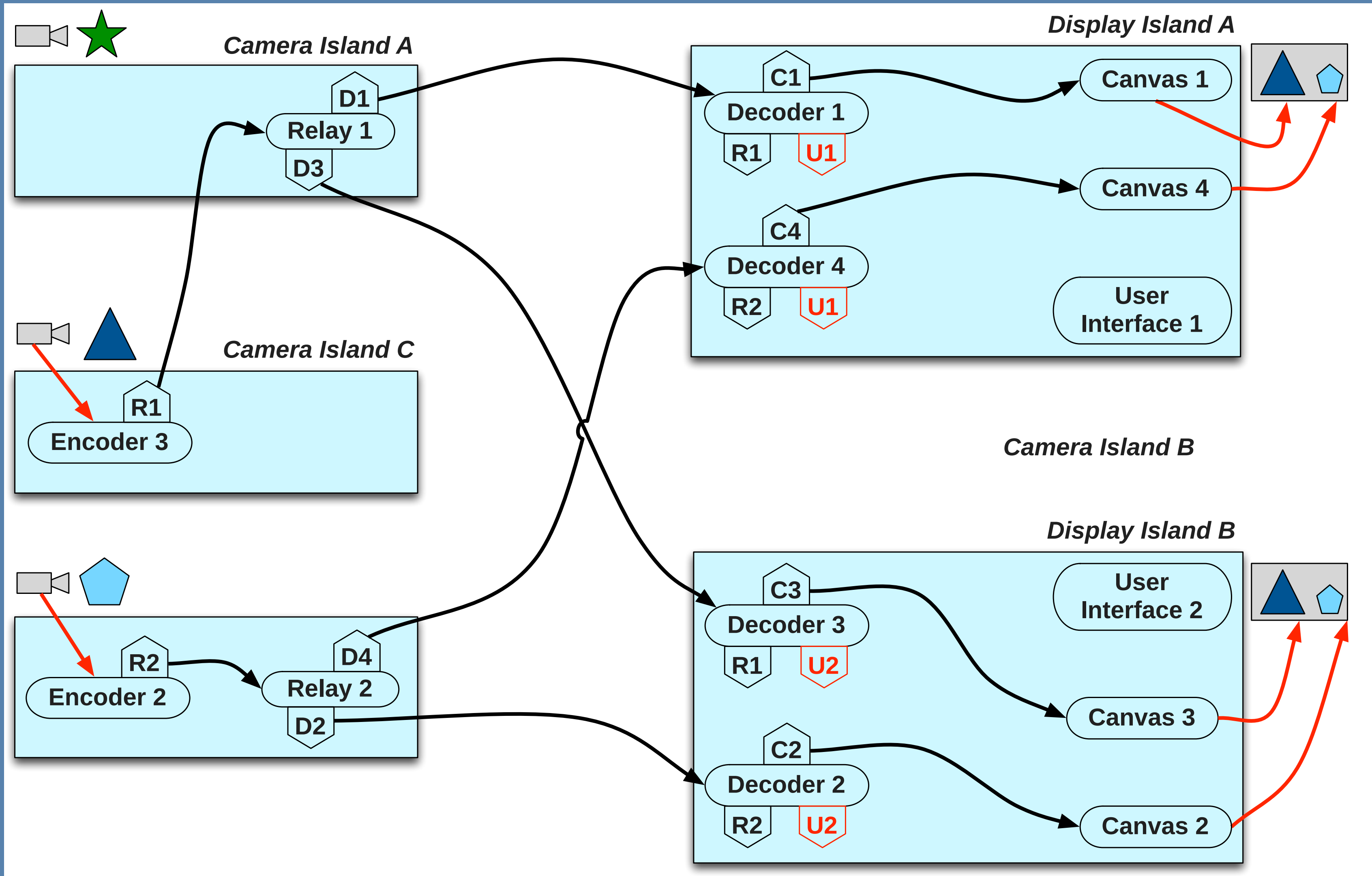
COASTcast The Movie: Sharing a Video



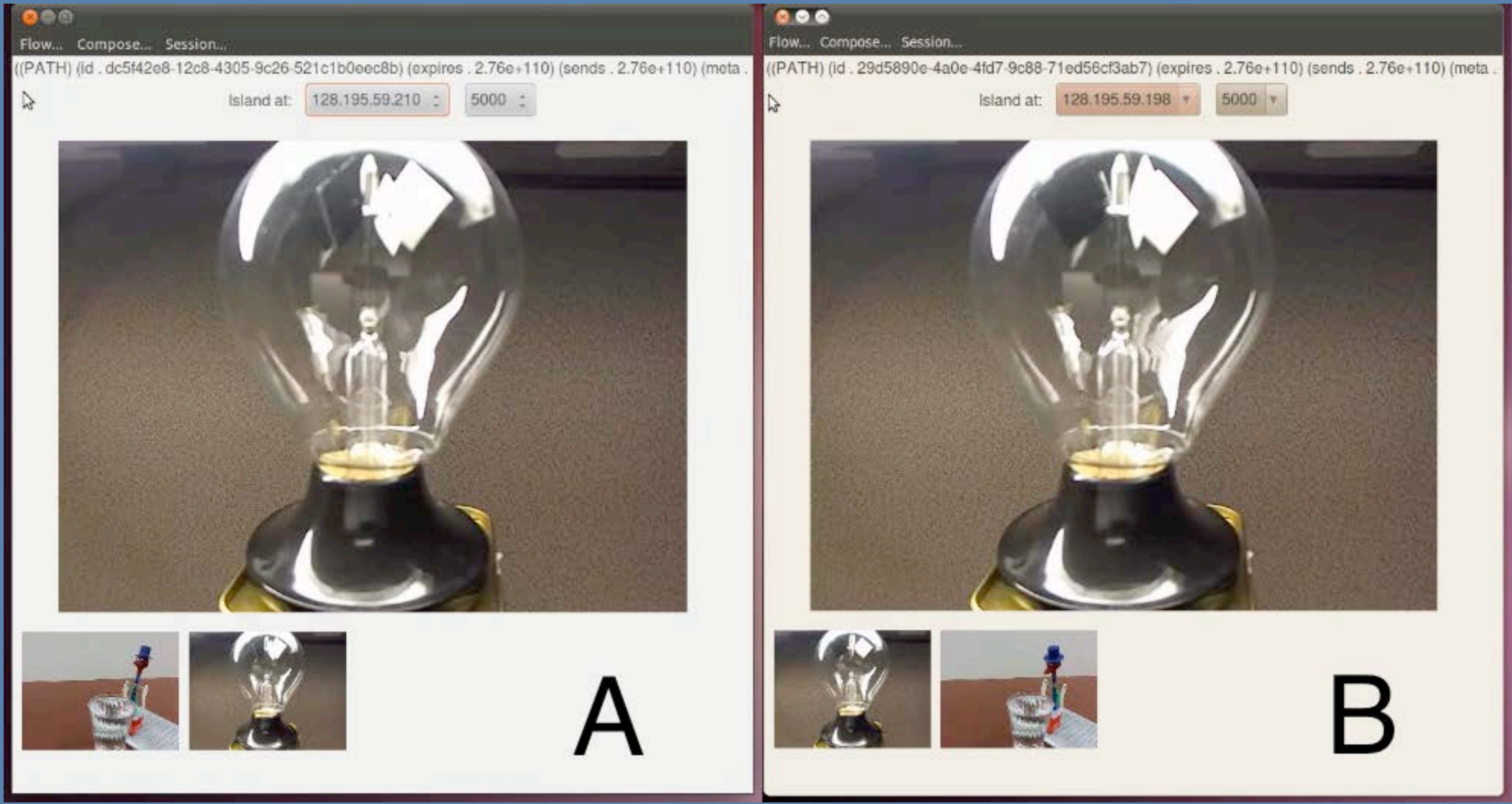
COASTcast: Moving a Video Source



COASTcast: Moving a Video Source



COASTcast The Movie: Change the Video Source



Related Work

- Capability Security
  - *Confused Deputy (Hardy, "The confused deputy: (or why capabilities might have been invented)," SIGOPS Operating Systems Review,1988)*
  - *Lambda calculus (Rees, "A security kernel based on the lambda calculus," PhD thesis, MIT, 1996)*
  - *Confinement (Shapiro, "EROS: A Capability System," PhD thesis, University of Pennsylvania, 1999)*
  - *Revocation & multi-level security (Miller & Shapario, Paradigm regained: Abstraction mechanisms for access control, ASIAN'03, 2003)*
  - *Object-Capability and Capability Languages (Miller, Robust composition: Towards a unified approach to access control and concurrency control, PhD thesis, John Hopkins University, 2006)*
  - *Non-delegation (Murray & Grove, "Non-delegatable authorities in capability systems," Journal of Computer Security, 2008)*
  - *Analytics (Murray, "Analysing the security properties of object- capability patterns," PhD thesis, University of Oxford, 2010)*
  - *Information flow control (Birgisson, Russo & Sabelfeld, "Capabilities for information flow," PLAS'11, 2011)*

Future Work/Summary

- Future Work
  - *Digital contract negotiation (Alegria Baquero)*
  - *Collaboration architectures for disaster response (Christoph Dorn)*
  - *Dynamic software update*
  - *Electronic health systems (emphasis on security and privacy)*
  - *Adaptive robotics*
- Summary
  - *Results suggest COAST is a step forward for decentralized applications*
    - Expressive (enough), efficient (enough) and secure (enough) for a variety of domains
  - *CURLs essential to robust COAST security*
  - *Mobile code is manageable given the tools of functional and communication capability*
  - *Architectural style can make significant contributions to application security*