Singularity Containers

Containers for …
HPC, analytics, machine learning, reproducible and trusted computing

http://sylabs.io

DAVE GODLOVE

- Software Engineer, Sylabs Inc.
- Previous:
  - Computational Biologist, National Institutes of Health, High Performance Computing Center
  - Postdoctoral Fellow, National Institute of Mental Health

GREGORY M. KURTZER

- CEO, Sylabs Inc.
- Senior Architect, RStor Inc.

- Previous:
  - HPC Systems Architect, LBNL (~20 years)

- Open Source Work:
  - Founder and project lead: Warewulf (2001)
  - Founder: Centos Linux (2003)
  - Founder and project lead: Singularity (2015)

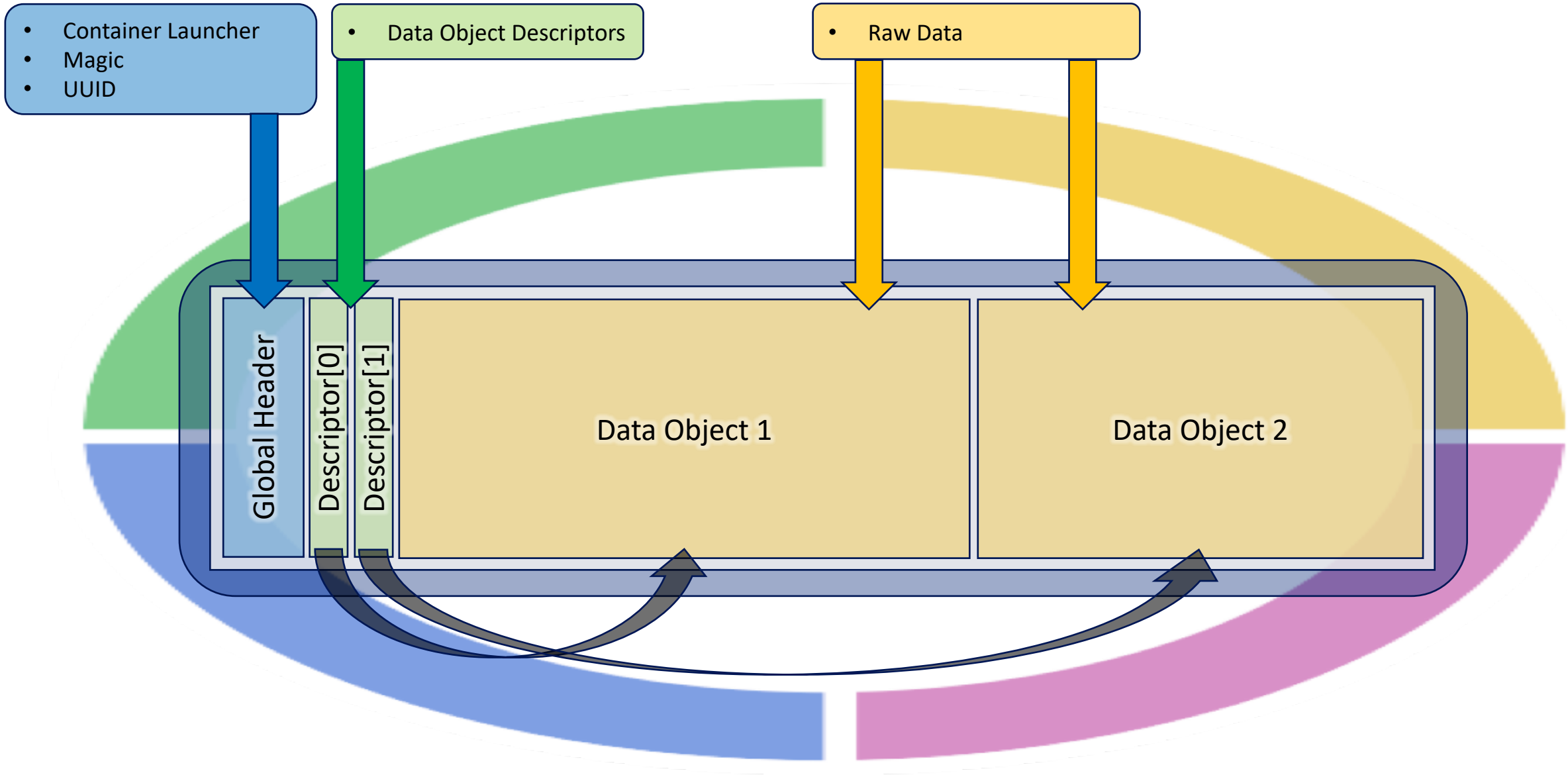
What makes Singularity different?

UNIQUE FEATURES OF SINGULARITY?

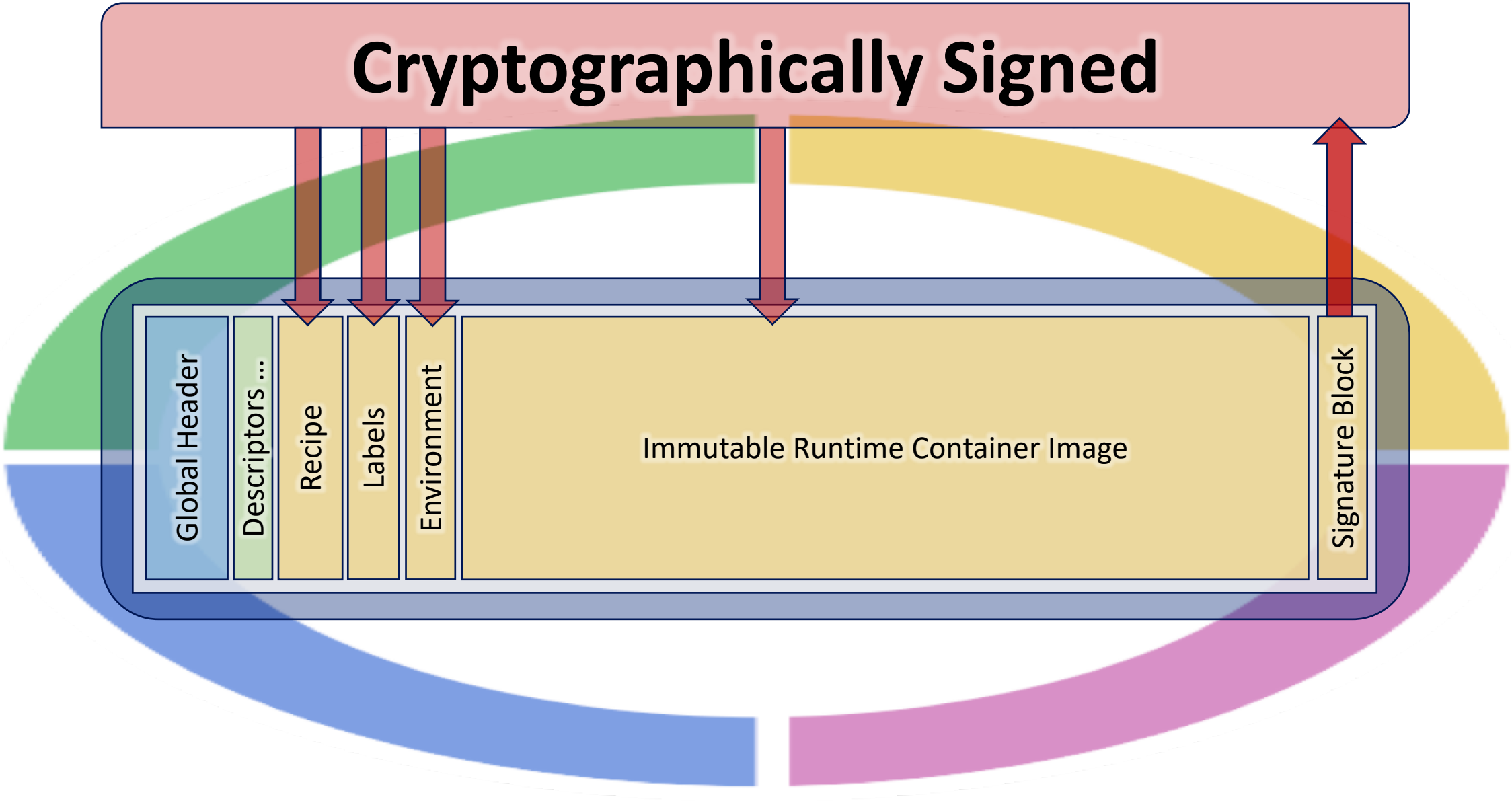
- **Built specifically to support HPC/Science:** response to HPC users' requests
- **Single file based container format:** verifiable via checksum and cryptographic signatures ensuring reproducible and validated software environments during runtime
- **Extreme mobility:** using standard tools (rsync, scp, GridFTP, NFS, etc.)
- **Controls compliant:** images can be easily archived and managed as any other data
- **Compatible:** with complicated architectures (e.g. MPI, IB, GPUs, etc.)
- **Security model:** designed to support untrusted users running untrusted containers *(rather then trusted users running trusted containers)*

The Singularity Image Format (SIF)

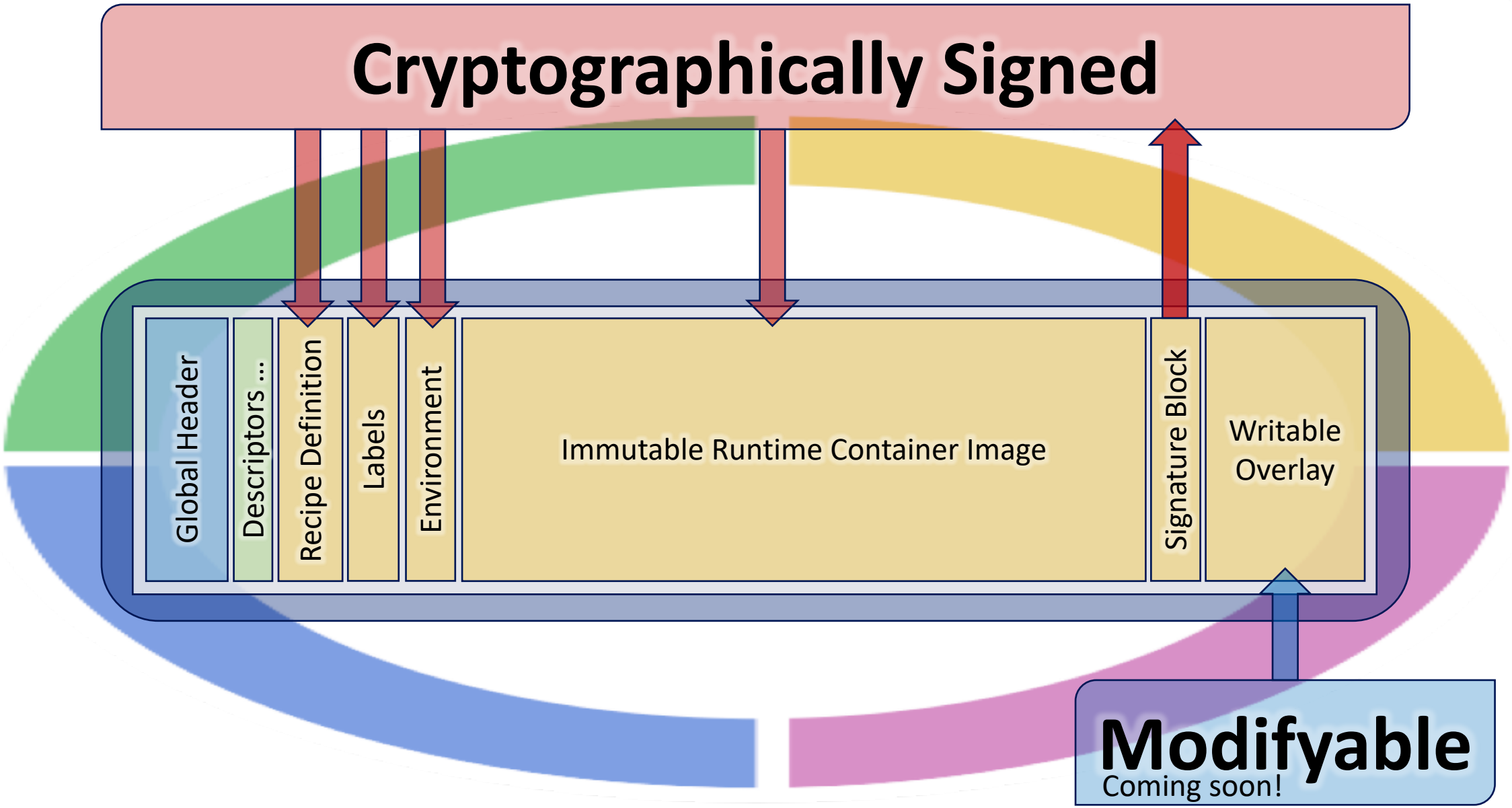
SINGULARITY IMAGE FORMAT (SIF)



SIF EXAMPLE: TRUSTED, SIGNED CONTAINERS



SIF EXAMPLE: TRUSTED, EVOLVING CONTAINERS



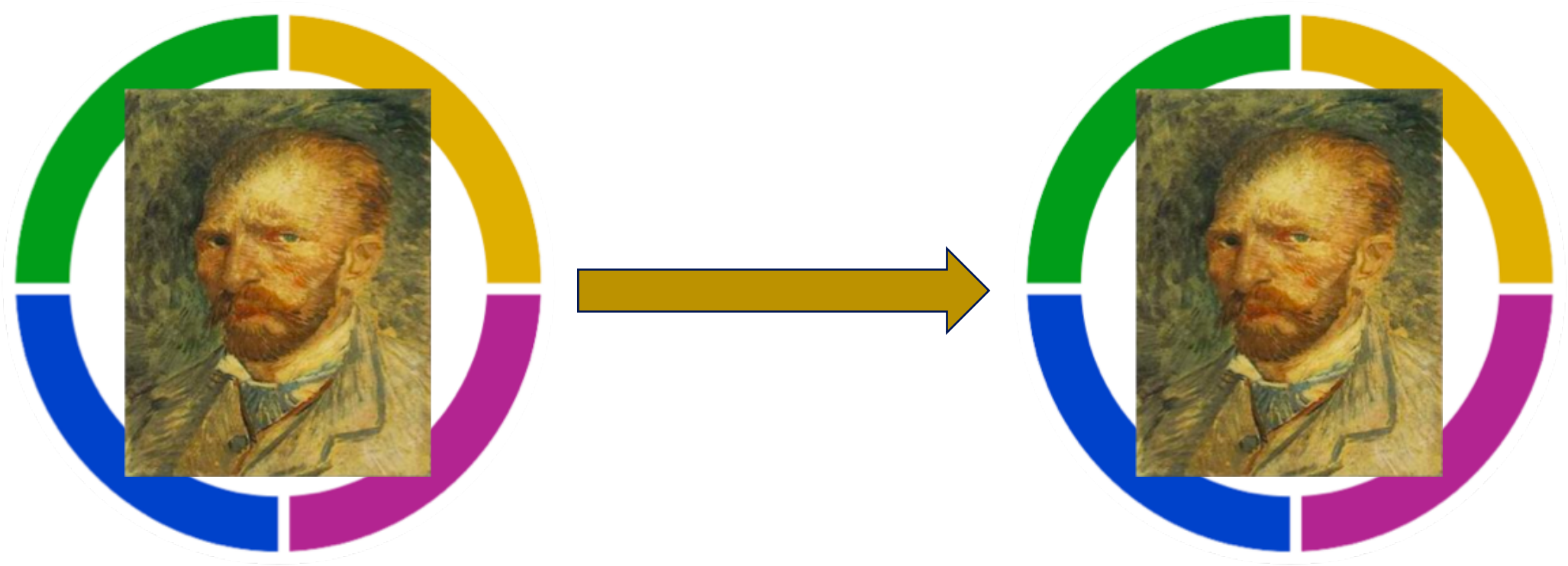
Reproducibility and Mobility

IS THIS “GOOD ENOUGH”?

=?

WHAT IS YOUR REPRODUCIBILITY PLAN?

BIT FOR BIT SOFTWARE REPRODUCIBILITY



SHA:
5f09a35a642a68c467bf230f5e5ea3218e4177a0

SHA:
5f09a35a642a68c467bf230f5e5ea3218e4177a0

ENTERPRISE

Microsoft Azure

amazon web services™

Google Cloud Platform

SINGULARITY ENABLES EXTREME MOBILITY

SCP, SFTP, GridFTP/Globus, Rsync, NFS, Lustre, Object Stores, etc..

Single file containers means that containers are easy to manage

Open Science Grid

XSEDE

Extreme Science and Engineering Discovery Environment

PENGUIN COMPUTING

COMPUTE

HPC Compatibility

HPC COMPATIBLE SECURITY MODEL

- **Base security assumption:** Untrusted users running untrusted containers
- **Limit user's potential security contexts:** We can not allow users to escalate to root, even in containers that they control (and know the root password to)
- **Allow user's access to data they own:** And limit access to data that they don't own

HPC COMPATIBLE CONTAINER ARCHITECTURE

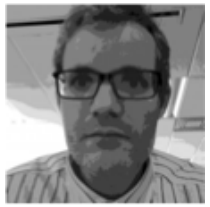
- **MPI support:** MPI jobs are also easily supported using a hybrid model
- **GPU:** Users are themselves within a container, and thanks to the Singularity security model, they can not escalate. This means we can share the GPU device into the container.
- **Resource managers:** Container processes are decedents of the RM (rather then a root owned container daemon which the RM has no direct control of)
- **Single file container images:** This image format is highly optimized for parallel FS

OPTIMIZED FOR PARALLEL FILE SYSTEMS

Objectives:

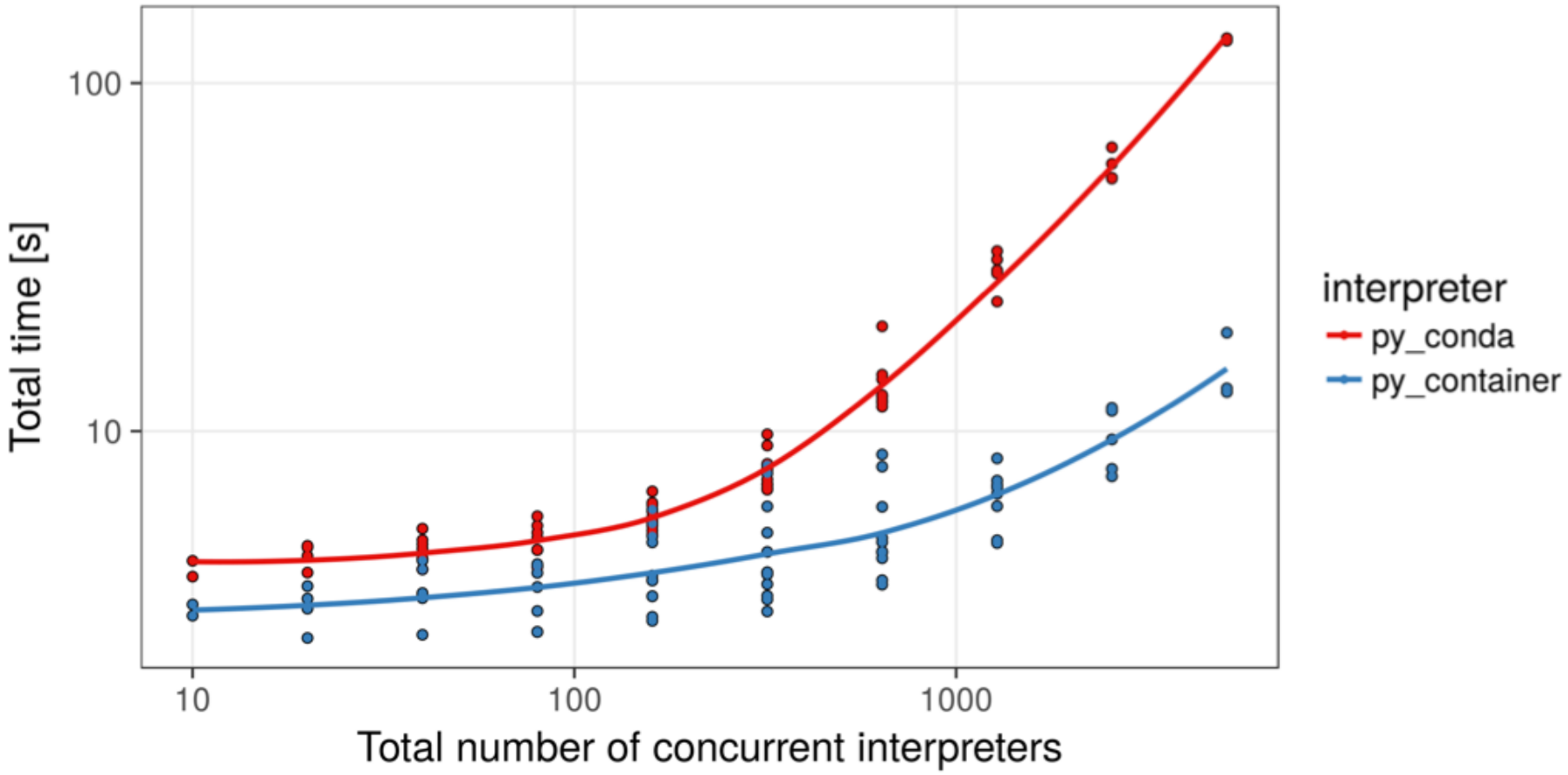
1. Measure scaling of python startup and import speed with increasing numbers of concurrent python interpreters
2. Compare scaling of a standard python installation with an identical containerized installation (Singularity).

*Note: Underlying file system is NFS, max jobs was 5120 over 320 nodes, graph is **logarithmic** on both axis.*



DR. WOLFGANG RESCH

HTTPS://GITHUB.COM/WRESCH/PYTHON_IMPORT_PROBLEM



Coming Soon

SOME ITEMS ON OUR ROADMAP

- Virtual container boot within instances
- Evolving signed containers
- OCI (Open Container Initiative) compliance
- Native Kubernetes support
- Performance profiling of contained applications
- Checkpoint / Restart of containers
- OS X and Windows support

SINGULARITY: CONTRIBUTORS

THANKS!

- http://www.sylabs.io/ The brand new home of Sylabs Inc
- http://singularity.lbl.gov/ Help docs for getting started
- https://singularity-hub.org/ CI build service for Singularity containers!
- https://singularity-tutorial.github.io/ Hands on Singularity tutorial for beginners.

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov