## Collaborative OPen Omics (COPO) -Building, Testing and Deployment



Email: ei.copo@earlham.ac.uk

Website: www.copo-project.org

Aaliyah Providence, Debby Ku, Felix Shaw

• COPO facilitates scientists to describe, store and access metadata using community standards and public repositories ensuring open data sharing.

• COPO employs modern technologies like Docker Swarm for orchestration, ensuring its availability, scalability, and security with separate networks and Docker secrets.



COPO services





Process (see figure below):

• A YAML file defines COPO services grouped in a Docker stack for easy management.



- Each COPO service is encapsulated in a Docker service.
- Docker services deployed to the swarm manager are scheduled as tasks on corresponding nodes (i.e. Node 1, Node 2 and Node 3), each invoking a container with a defined Docker image.
- Containers within the same network communicate via assigned IP addresses.
- COPO securely stores sensitive information as Docker secrets.
- COPO services attach to volumes to ensure data persistence during restarts.
- Some COPO services share storage by attaching to the same volume.









3

- COPO is a metadata brokering platform that utilises GitHub for code storage and versioning, Docker for its deployment and CircleCI for its testing.
- The COPO system consists of multiple services via Docker stack that is managed by a Docker swarm manager and grouped across several nodes.



**Decoding Living Systems** 

Earlham Institute, Norwich Research Park, Norwich, Norfolk, NR4 7UZ, UK

www.earlham.ac.uk







Acknowledgements

This work was supported by the Biotechnology and Biological Sciences Research Council (BBSRC), part of UK Research and Innovation, through the Cellular Genomics (CELLGEN) Grant BBS/E/ER/230001A, Data Infrastructure and Algorithms Group Grant EI-G-Data, Decoding Biodiversity (DECODE): Development of Novel Experimental and Bioinformatic Tools for Genomic Diversity and Analysis Grant BBS/E/ER/230002A and EI - Darwin Tree of Life Grant GP182 at the







