

# The L-functions and Modular Forms Database (LMFDB)

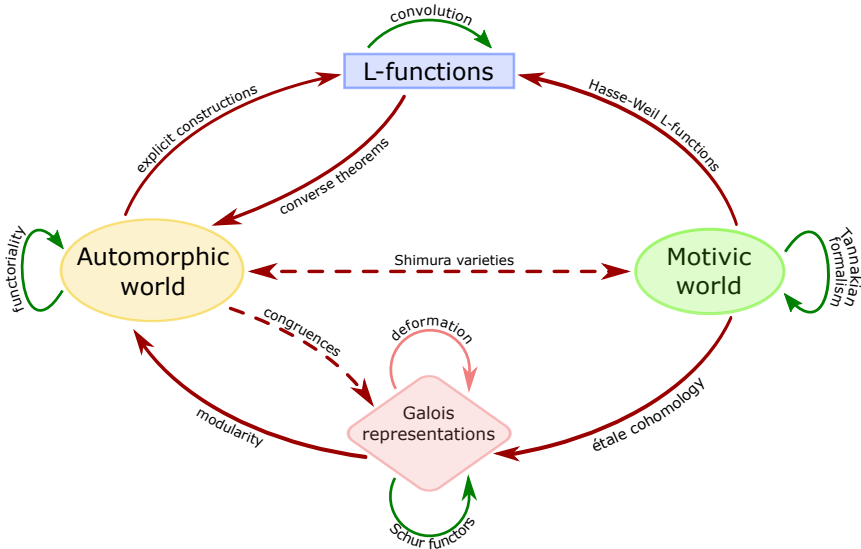
Andrew Sutherland

Massachusetts Institute of Technology

International Congress on Mathematical Software 2020

July 13, 2020

# The LMFDB universe



## Motivation and desiderata

- Number theory has long been, in part, an experimental science. Data is often the source of conjectures that lead to theorems.
- Exhaustive enumeration allows one to prove theorems and exposes holes (both in theory and in implementations) by finding all the special cases.
- The database should be easily accessible and comprehensible to as broad an audience as possible, serving both novices and experts.
- All data should have a clear and citable provenance: how it was computed, by whom, to what precision, and under what assumptions, if any.
- Search and aggregation tools are needed to maximize the utility of the data.

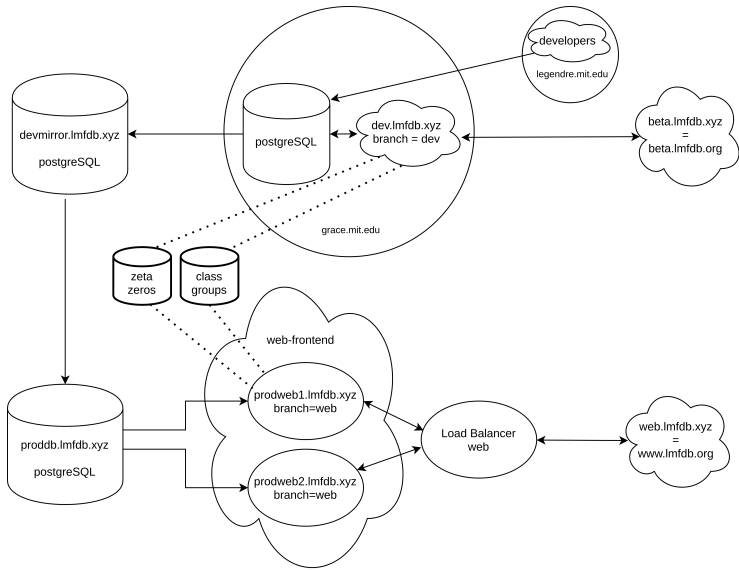
## History and organization

- The LMFDB was first conceived at an AIM workshop in 2007. The first production release was launched in 2016.
- More than 100 researchers have contributed data, code, or both. All LMFDB software and the technology stack that supports it is open source.
- We have received substantial financial support from EPSRC, NSF, and the Simons Foundation, as well as smaller grants from many other sources.
- The LMFDB is overseen by a board of managing and associate editors.
- The LMFDB is currently hosted on servers running on Google Cloud Platform. A full mirror and a complete development configuration is hosted at MIT.
- In 2018 we migrated the entire database from MongoDB to PostgreSQL.

## Technical specs

- 1TB zeta zeros and 2TB class group data available for download.  
1TB of searchable data in 800+ million rows of 100+ PostgreSQL tables.  
The average row is about 1KB but many are well over 100KB.
- There are two load-balanced webservers and a database server on GCP.  
A mirror of the development database serves as a staging area for updates.
- We use the Flask web framework running on Python 3 using SageMath.  
All database access occurs via a custom interface layer built on psycopg2.
- Software development and version control is coordinated via GitHub.  
Our repository is public and the code is available under a GPL 2+ license.
- All software and data changes are extensively reviewed on the development site before they are put into production.

# Network configuration



## Some challenges

- Data sets whose objects vary widely in complexity.
- Schema flexibility versus performance.
- Permanently labeling mathematical objects.
- Completeness and scope.
- On-demand computation.
- Rigor, reliability, reproducibility.