Modeling Workgroup Meeting Quarterly Review

Optimization Update: Development of Efficient MO Procedures

Gregorio Toscano, Kalyanmoy Deb, Pouyan Nejadhashemi, Vahid Rafiei, and Julian Blank

APRIL 2022

MICHIGAN STATE UNIVERSITY

Previous Presentations (Jan 2022 and Oct, 2021)

- Overview of the developed web interface
- Short demo of the web interface approach
- (Completed) Objective 1: Understanding the CAST system and Development of an Efficient Single-objective Hybrid Optimization Procedure

Agenda

- Objective 2: Development of Efficient Multi-objective Optimization Procedures
  - Oct 1, 2021 to September 30, 2023 (24 months)
- Current Accomplishments:

  1) Submission of article: Gregorio Toscano, Juan Hernández, Julian Blank, Pouyan Nejadhashemi, Kalyanmoy Deb, and Lewis Linker. *Large-scale Multi-objective Optimization for Water Quality in Chesapeake Bay Watershed* to WCCI 2022 IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE (Decision awaited)

     A. Develop of a generative strategy (Opt7) in which each optimal solution is found one at a time by using the epsilon-constraint method.

     B. Develop of a simultaneous strategy (Op8) in which multiple optimal solutions are found in one optimization run by using NSGA-III.

  2) Containerization of the user interface developed on Objective 1.

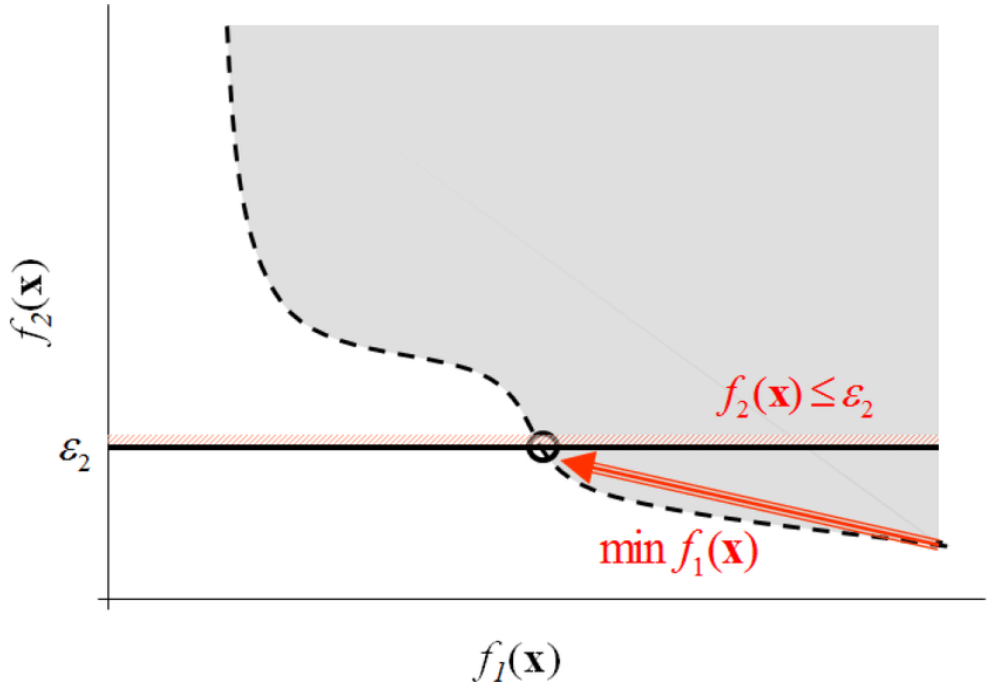
  3) Development of microservices (Opt4Cast APIs).

Experiments Performed

- Experiment 1: Knowledge incorporation through solution injection.
- Experiment 2: Reduction of constraints with a repair approach.
- Experiment 3: Scale-up study.
- Experiment 4: Deciphering common patterns of BMP allocation in final trade-off solutions.

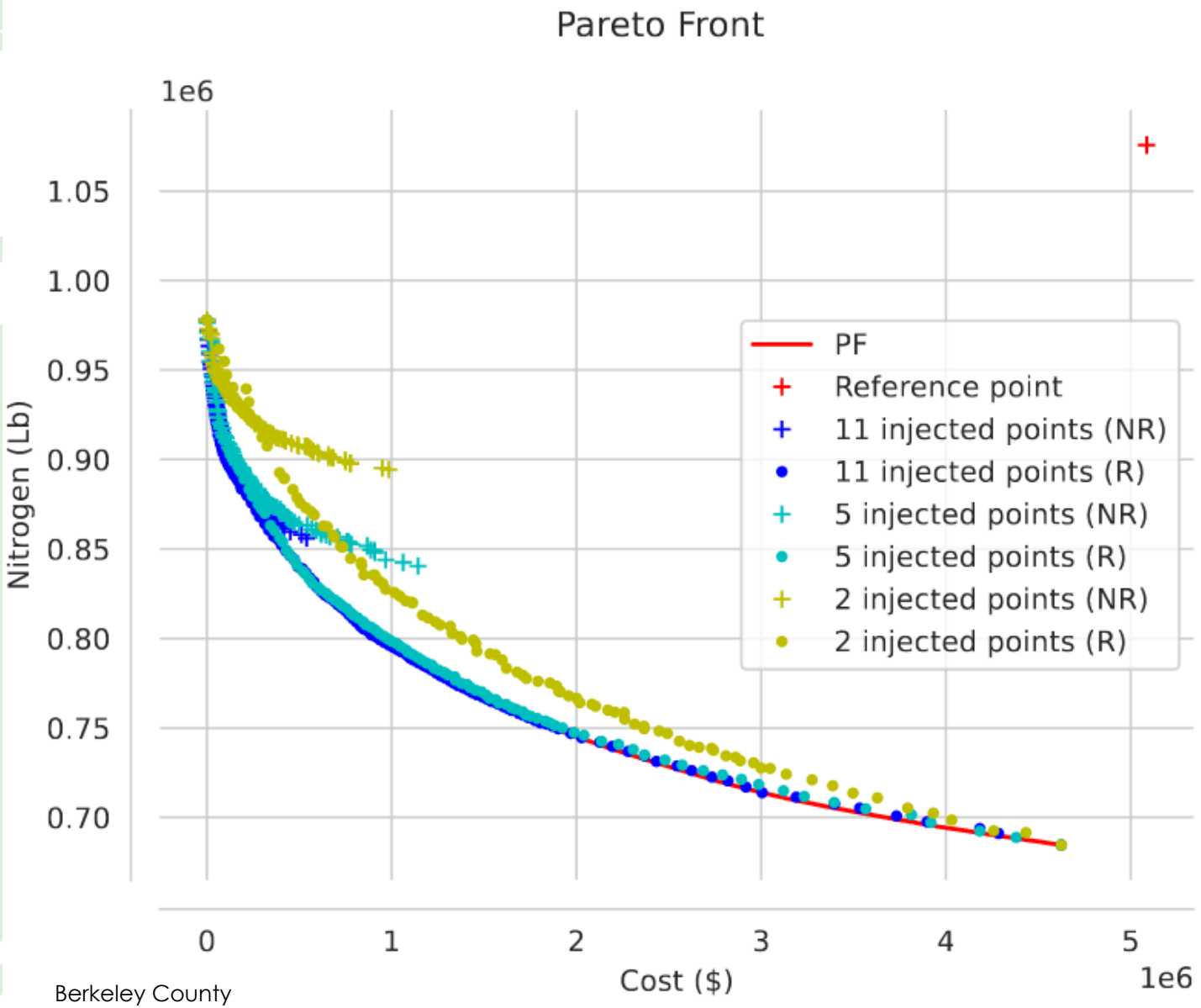
Epsilon constraint

- We minimize cost, while we set a certain epsilon as a goal for the method.
- Although it is effective, it is not efficient as it only provides one single solution per execution.
- The final solution of this approach will be injected into our multi-objective optimization approach.
- We want to inject as fewer solutions as possible.



Knowledge incorporation + Repair approach

- Injected points are important for evolutionary algorithm convergence
- Linear constraints are repaired to make population members feasible.

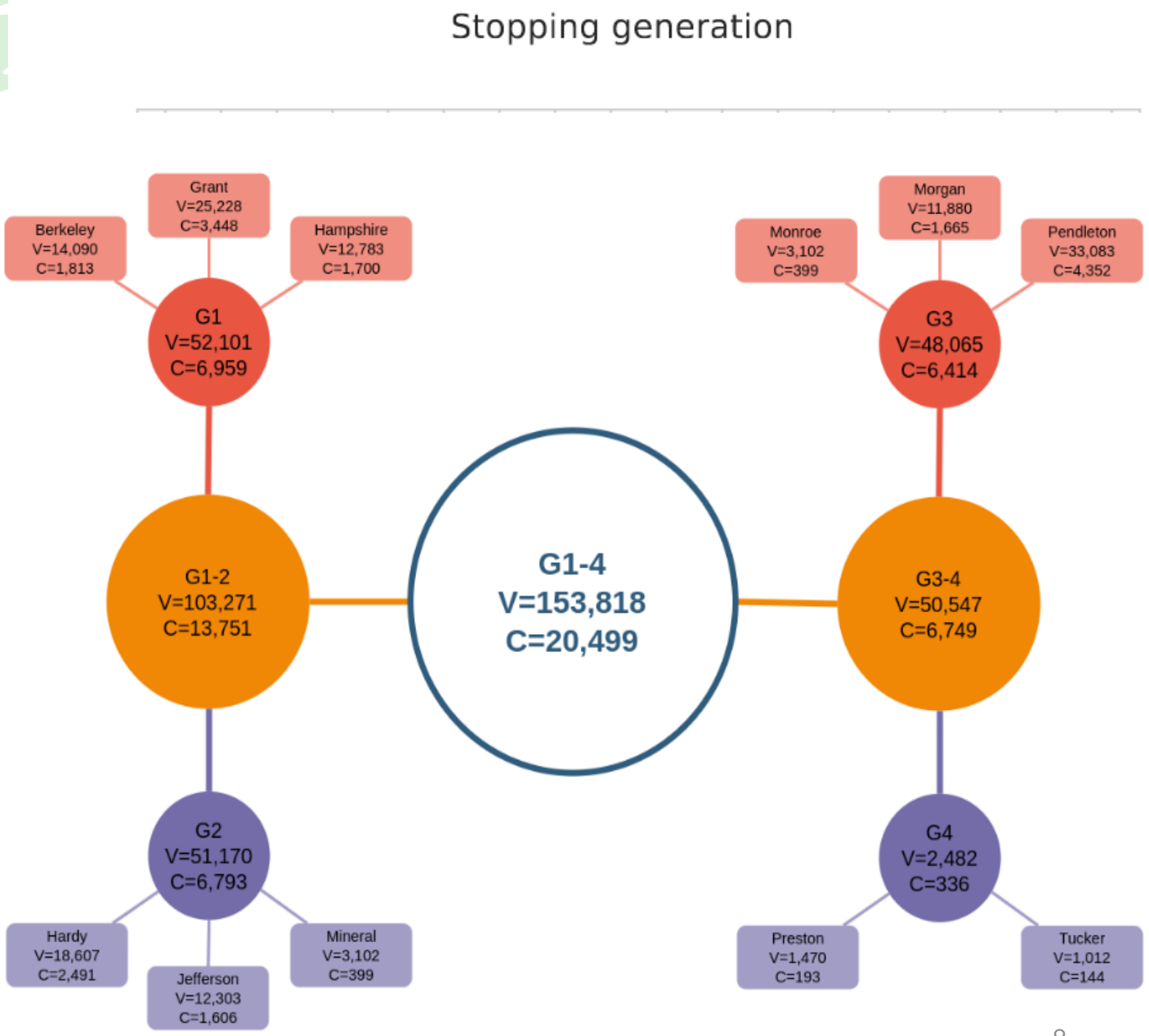


Area of study: West Virginia
Problem: Large-scale, Highly Constrained

County	#Variables	#Constraints	Base N_2 (lbs)
Berkeley	14,090	1,813	977,896
Grant	25,228	3,448	1,049,450
Hampshire	12,783	1,700	1,012,797
Hardy	18,607	2,491	1,344,295
Jefferson	12,303	1,606	1,018,012
Mineral	20,260	2,698	763,864
Monroe	3,102	399	48,655
Morgan	11,880	1,665	271,134
Pendleton	33,083	4,352	1,133,327
Preston	1,470	193	4,683
Tucker	1,012	144	1,702
Total	153,818	20,509	7,625,818

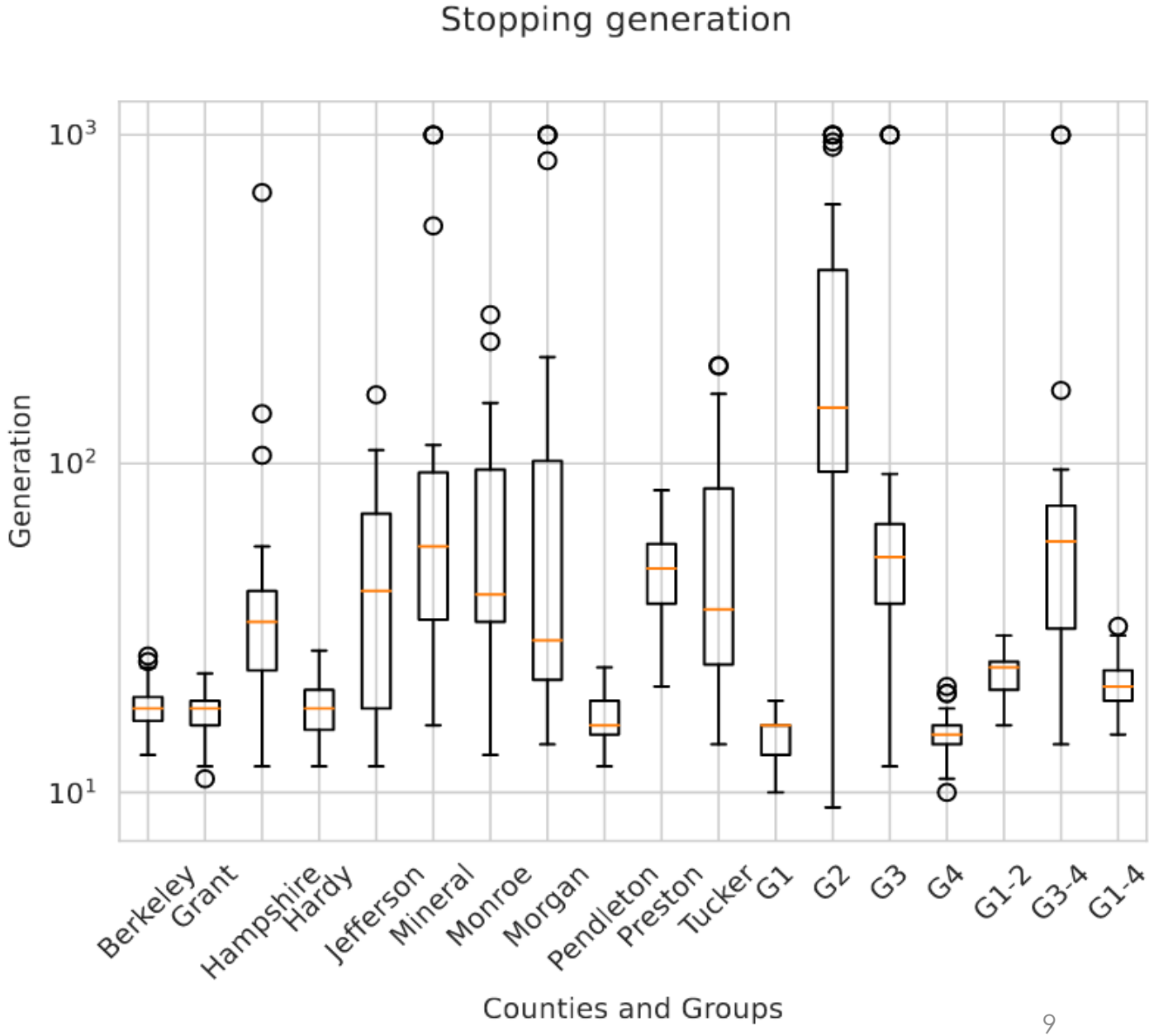
Scale-up Study

- Optimize each county independently
- Solve small aggregated problem (groups of 3 counties at most)
- Solve medium aggregated problem (groups of 6 counties at most)
- Solve the entire state

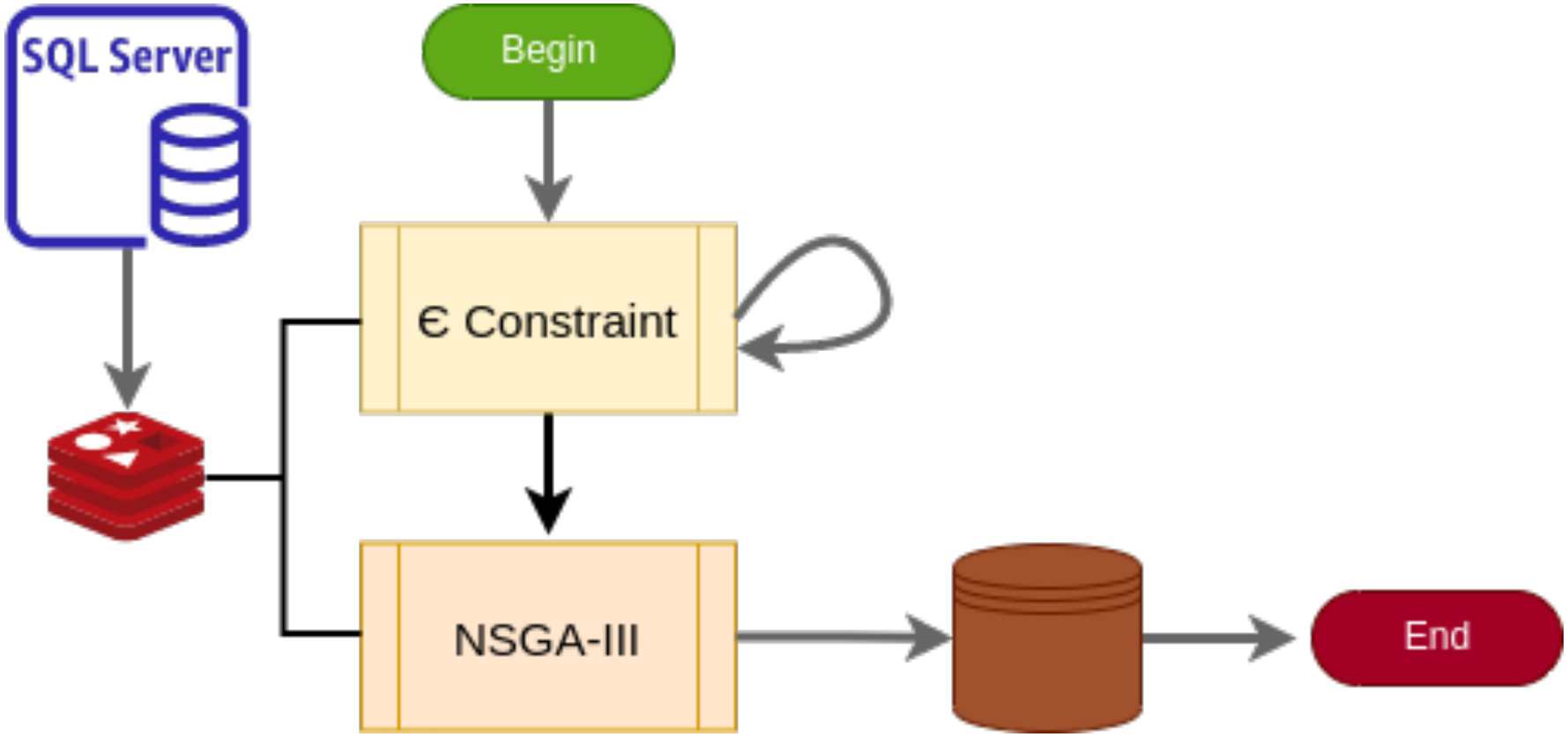


Interesting Results

- Grouping of counties for optimization finds solutions faster than optimizing independently.
- Proposed Customized NSGA-III's total evaluations to converge does not increase with the addition of counties.
- The final solutions use only 10% or less available BMP options to produce a cost-nitrogen trade-off for most counties.



Developed Algorithm



Containerization

- What is it missing?: time execution measuring.
- Why? The optimization approach was not running on the same infrastructure than CoreCAST. Therefore, networking time would lead to misreads.
- How to solve it? Running the optimization approach in the same infrastructure.
- Solution: Create a container image with all the dependencies and run it in the same infrastructure

Microservices

Monolith

- Server-side system based on single application
- Easy to develop deploy manage

Challenge

- Highly dependent
- Language/framework
- Growth
- Scaling

VS

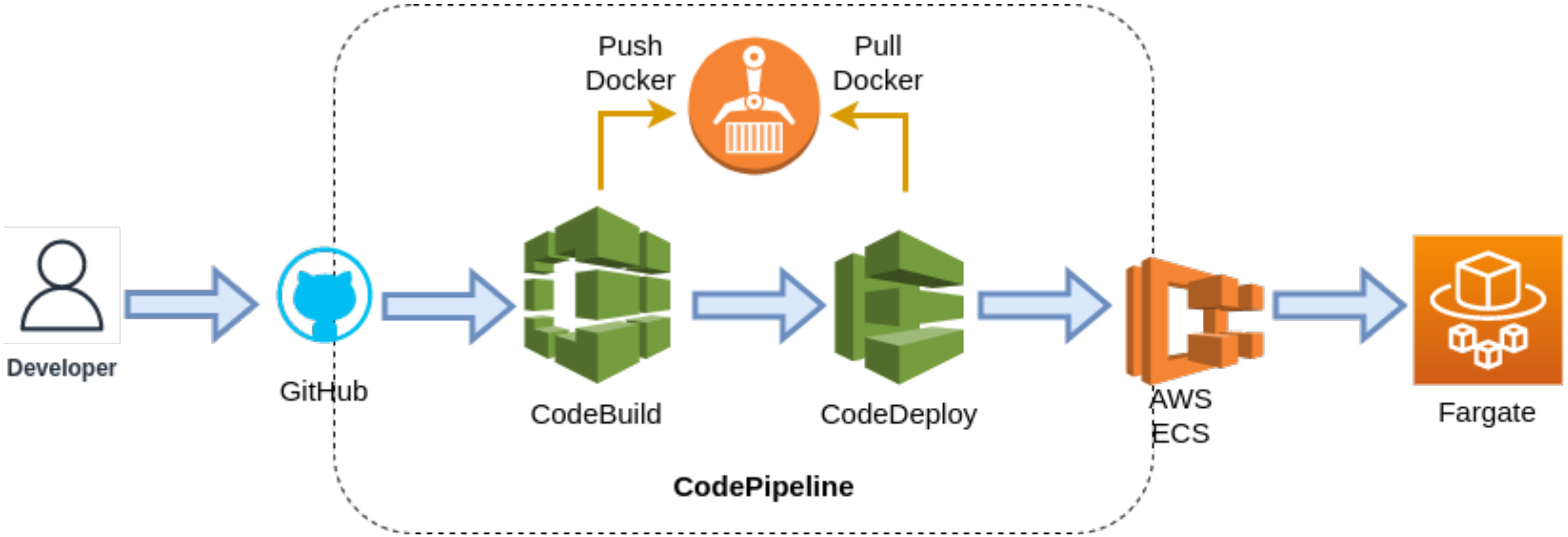
Microservices

- Every app function is its own service
- Own container
- Commmunicate via APIs

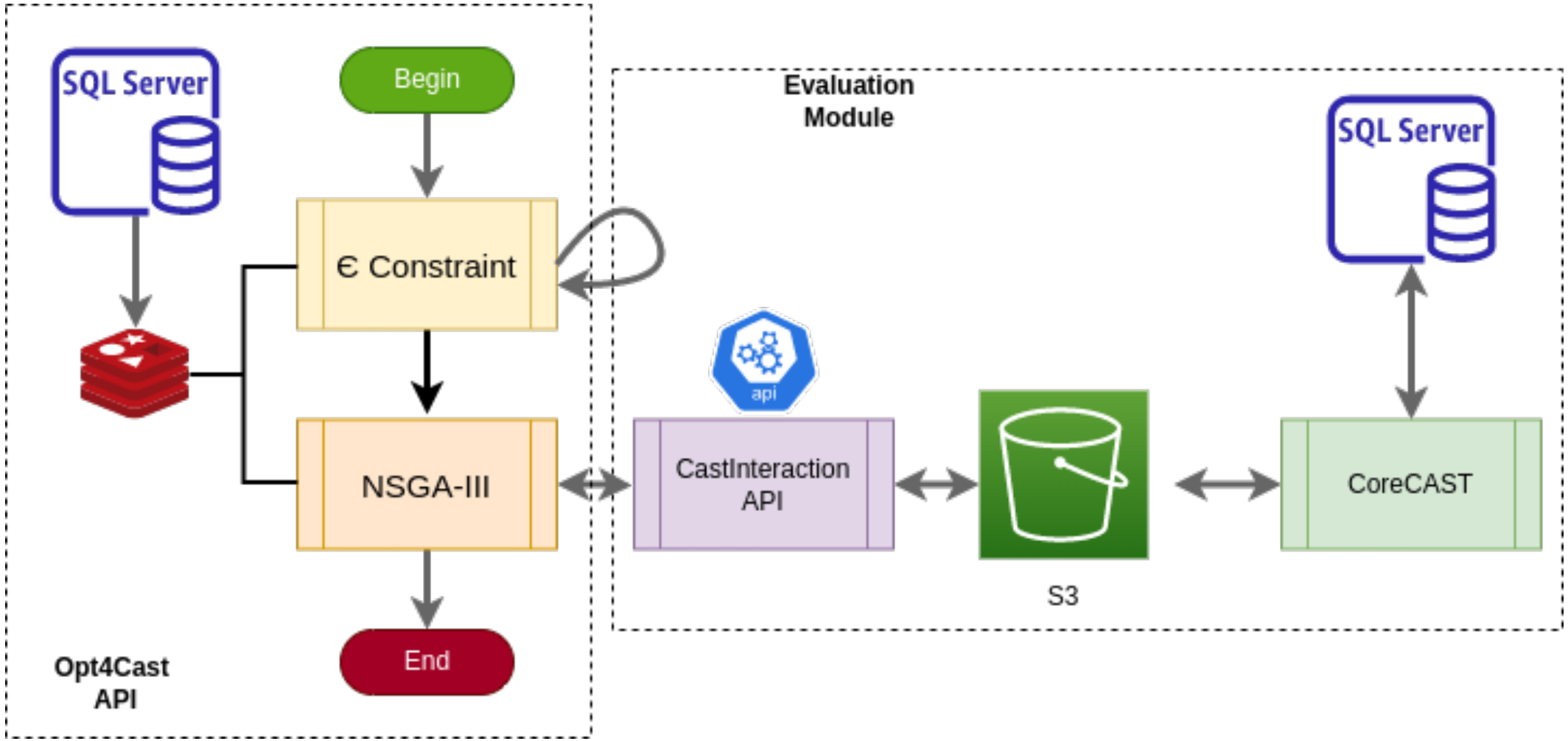
Advantage

- Language independent
- Iterate at will/dev ops pipeline
- Less risk in change
- Independent scaling

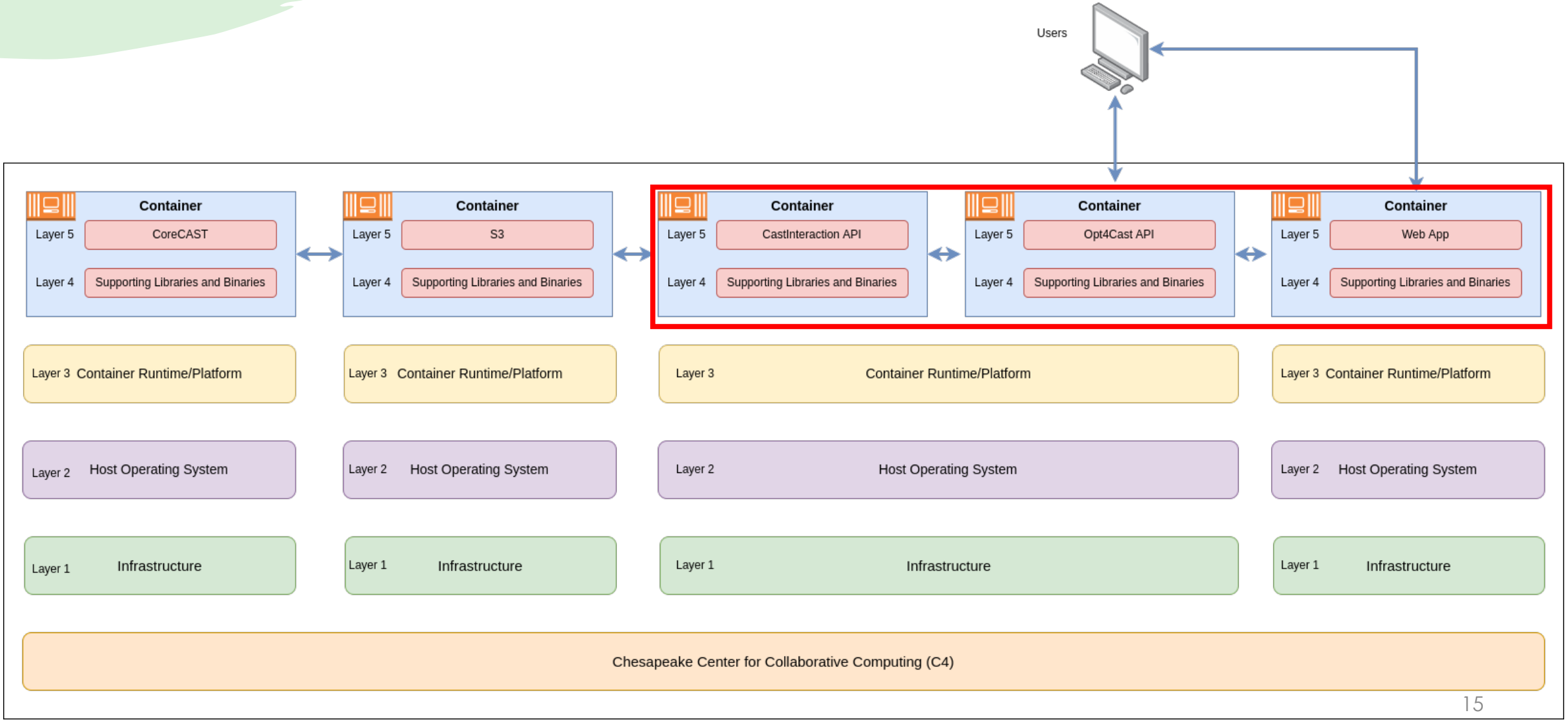
Code Pipeline



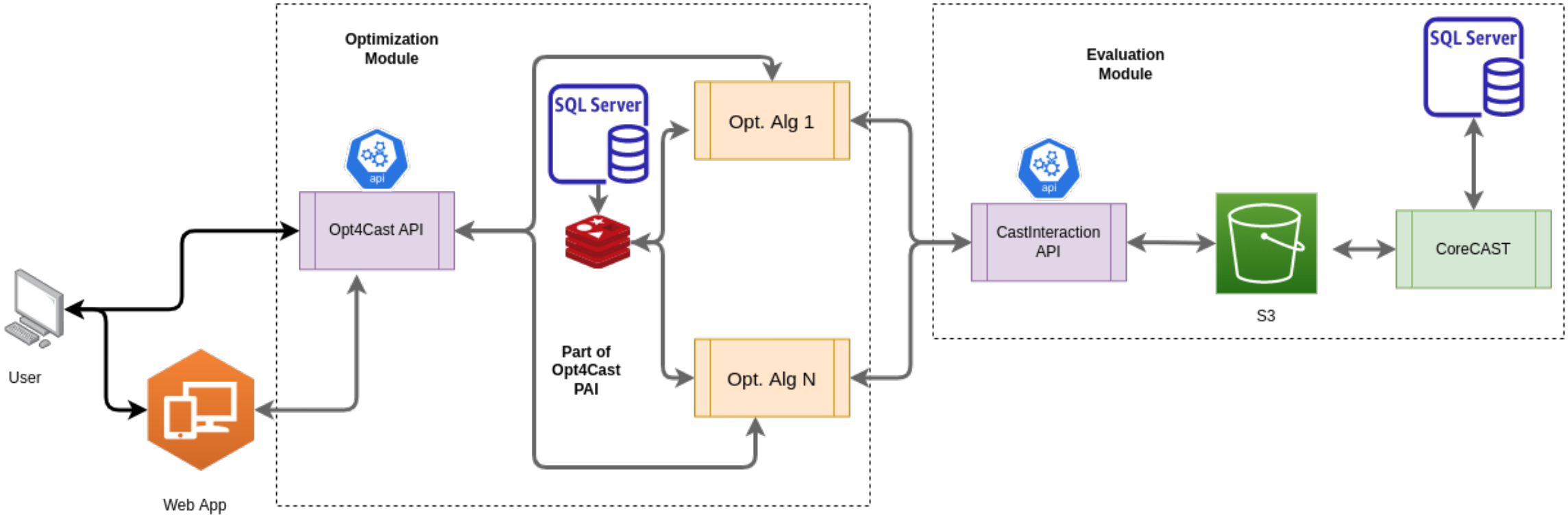
NSGA-III + CoreCAST (under development)

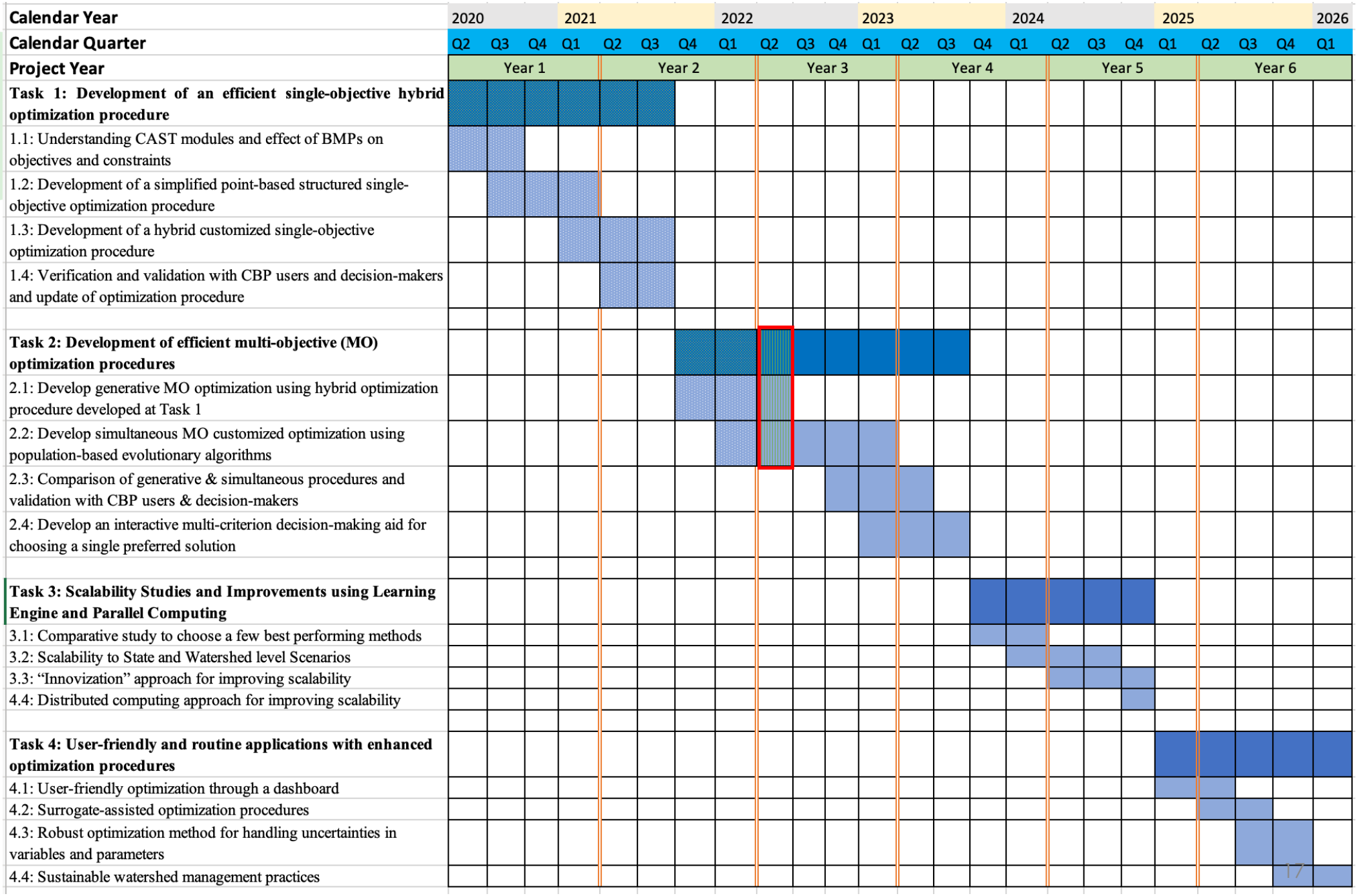


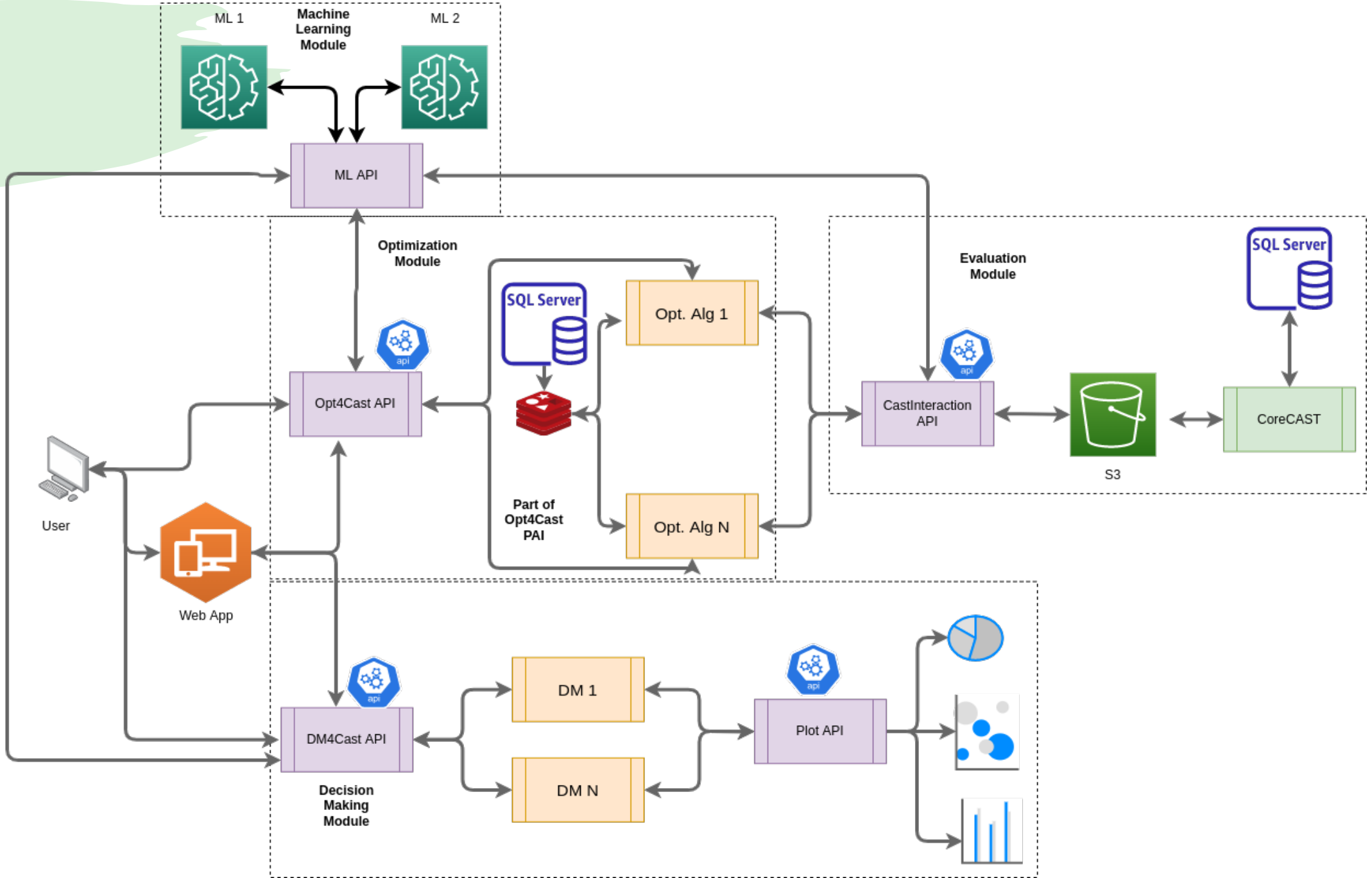
Current Scheme (under development)



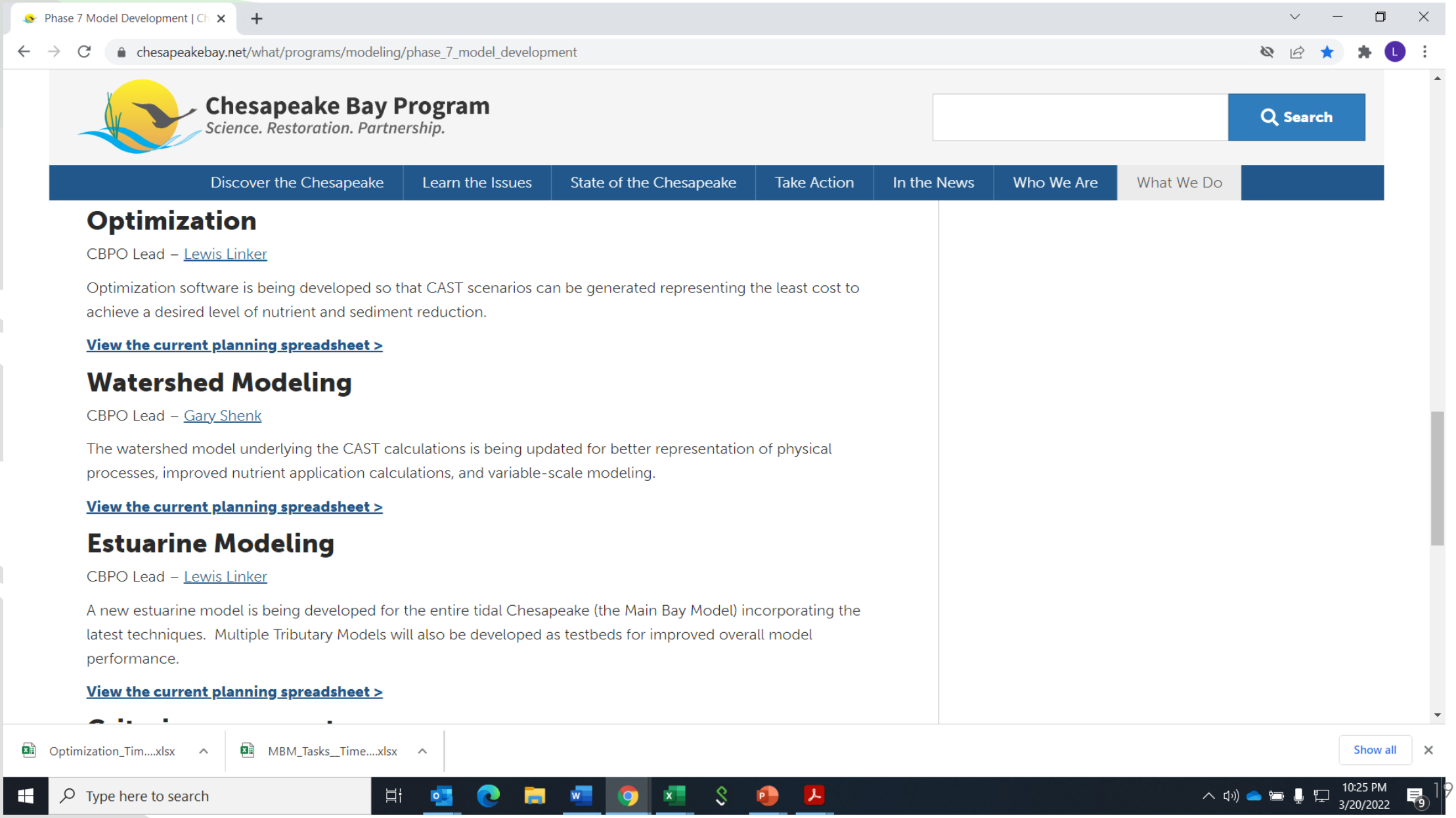
Opt4Cast Microservices (under development)




Calendar Year
2020
2021
2022
2023
2024
2025
2026
Calendar Quarter
Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1
Project Year
Year 1
Year 2
Year 3
Year 4
Year 5
Year 6
Task 1: Development of an efficient single-objective hybrid optimization procedure
1.1: Understanding CAST modules and effect of BMPs on objectives and constraints
1.2: Development of a simplified point-based structured single-objective optimization procedure
1.3: Development of a hybrid customized single-objective optimization procedure
1.4: Verification and validation with CBP users and decision-makers and update of optimization procedure
Task 2: Development of efficient multi-objective (MO) optimization procedures
2.1: Develop generative MO optimization using hybrid optimization procedure developed at Task 1
2.2: Develop simultaneous MO customized optimization using population-based evolutionary algorithms
2.3: Comparison of generative & simultaneous procedures and validation with CBP users & decision-makers
2.4: Develop an interactive multi-criterion decision-making aid for choosing a single preferred solution
Task 3: Scalability Studies and Improvements using Learning Engine and Parallel Computing
3.1: Comparative study to choose a few best performing methods
3.2: Scalability to State and Watershed level Scenarios
3.3: "Innovization" approach for improving scalability
4.4: Distributed computing approach for improving scalability
Task 4: User-friendly and routine applications with enhanced optimization procedures
4.1: User-friendly optimization through a dashboard
4.2: Surrogate-assisted optimization procedures
4.3: Robust optimization method for handling uncertainties in variables and parameters
4.4: Sustainable watershed management practices


ML 1
Machine Learning Module
ML 2
ML API
Optimization Module
SQL Server
Opt. Alg 1
Opt4Cast API
Part of Opt4Cast PAI
Opt. Alg N
Evaluation Module
SQL Server
CastInteraction API
S3
CoreCAST
User
Web App
DM4Cast API
Decision Making Module
DM 1
DM N
Plot API

https://www.chesapeakebay.net/what/programs/modeling/phase_7_model_development



Thank you