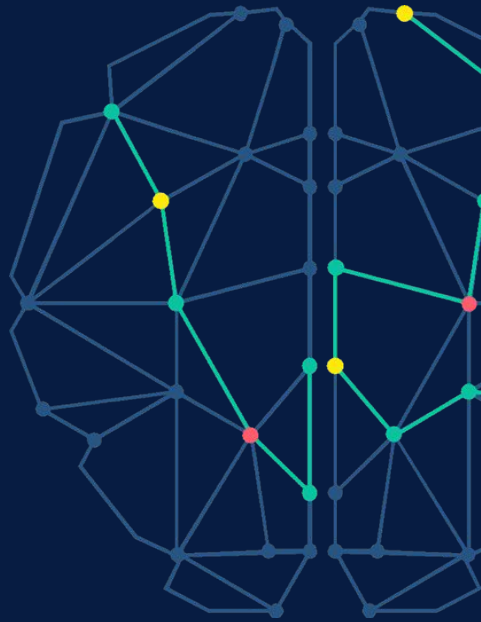
# SeaLights

Test Optimization Training



# Agenda

Sealights

# Agenda

Sealights

# Agenda

Sealights

# SeaLights TIA Use Cases

Sealights

## Reduce Cost

**Reduce Infrastructure cost**

You must spin up more cloud machines as you add more tests to your pipelines.
By selecting and running only the most relevant tests per each build or release you can reduce cloud and infrastructure costs significantly

## Reduce Time

**Shortcut Release time frame**

test smarter by selecting and running only the most relevant tests per each build or release, cutting significant of your testing cycle time.

## Reduce Troubleshooting

**Reduce Fail Tests**

Reduce costs by executing only the impacted
tests, cutting irrelevant failed tests troubleshooting costs, and
reducing QA and Engineering resources

# How it works

Sealights

Sealights TIA is a solution which creates a correlation between tests and methods that are invoked at the same time.

The linkage can be created a few ways:

1. Statistical model based on time correlation (Test time and code execution time)
2. Statistical + detemenitic model (Calibration) - improve the test to code mapping model by running the statistical model + the ability to lock code to test association for tests that are running sequential
3. Deterministic model - for specific type of tests like unit tests or in certain supported language and testing FW - a shared context is gluing test and code elements (powered by Open telemetry)

Based on the correlation of tests to code, if tests were not impacted by the code modifications, SeaLights can recommend tests to be skipped

Sealights also support various policies to affect how strict the recommendations are when deciding which tests can be skipped

# How it works

Sealights

Sealights Code to test mapping is designed in a way that all the both methods that belong to a test and some that don’t are captured and linked to it.

This means that some tests that do not need to run will still be recommended to run, but **ensures** that tests that need to run will **always** run.

Improvements can be made over time without an impact to quality

# Terms

Sealights

| | |
|---|---|
| Statistical based execution | SeaLights monitors the methods hit during the time tests are run in parallel.<br>Methods that belong to a test will have a high percentage of hits, whereas methods that do not belong to the tests and are hit sporadically will have a lower percentage of hits.<br>Based on the percentage, Sealights will consider a method hit or not hit by a test |
| Calibrated execution | Running tests on their own (in sequence) will allow Sealights to understand better which methods were actually hit by the tests with less noise.<br>When doing so, you can mark the run as calibrated, and Sealights will give this information priority over non-calibrated runs |
| Test to code mapping | The mapping of tests to code based on the methods hit during the time the tests were run |
| Recommendation | A list of tests to skip based on if they were impacted by code changes, previously failed or previously recommended and not run. |
| Previously Fail Test | A test that failed will by default be recommended to run even if the code changes did not impact it.<br>This policy can be changed to not recommend failed tests. |

# Configuration Options:

Sealights

## Statistical Model

Testing Strategy: Parallel Execution

**What:**
Code to test mapping is updated by the percentage of times a method was hit by a test during each time it ran.
Higher percentage methods hit will be considered to belong to a test
**Why:**
Minimum amount of change to the customers processes and test runs
**Value:**
Value should improve over time as the statistical information of the methods hit by tests are improved
**Trade Off:**
It takes a number of runs till good savings are provided, and in some cases based on how tests run, good saving might be difficult to achieve

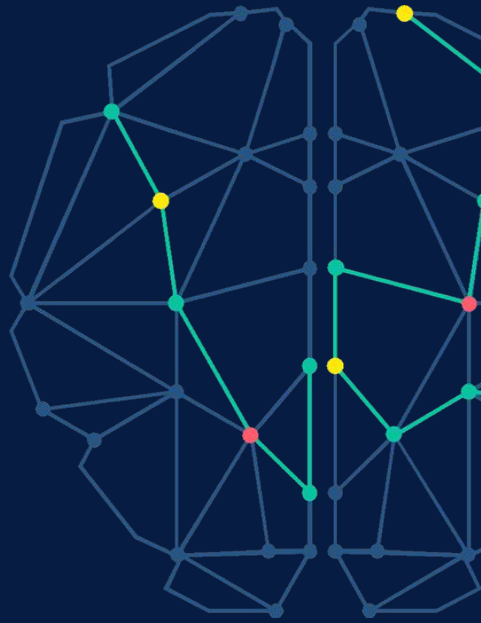
## Statistical + Calibration Model

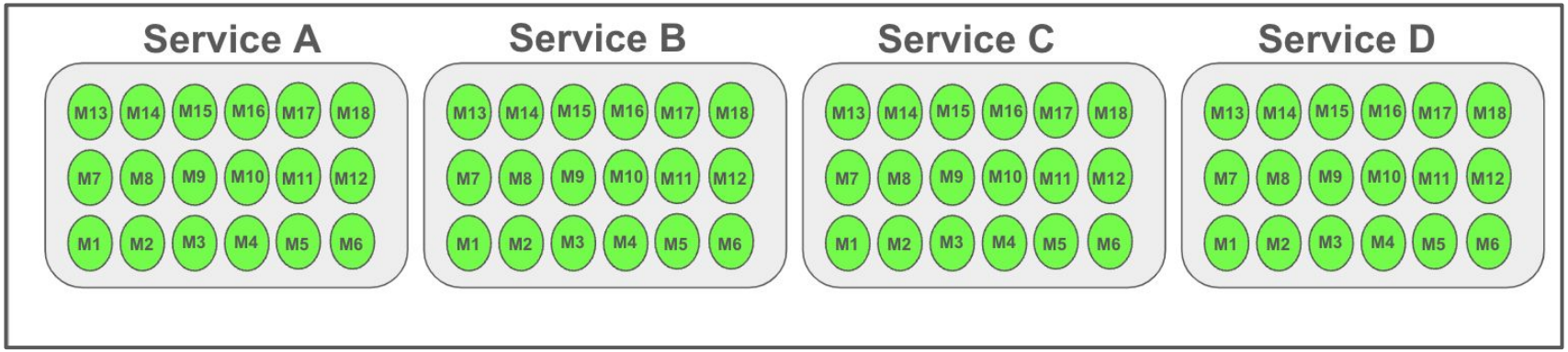
Testing Strategy: Sequential Execution

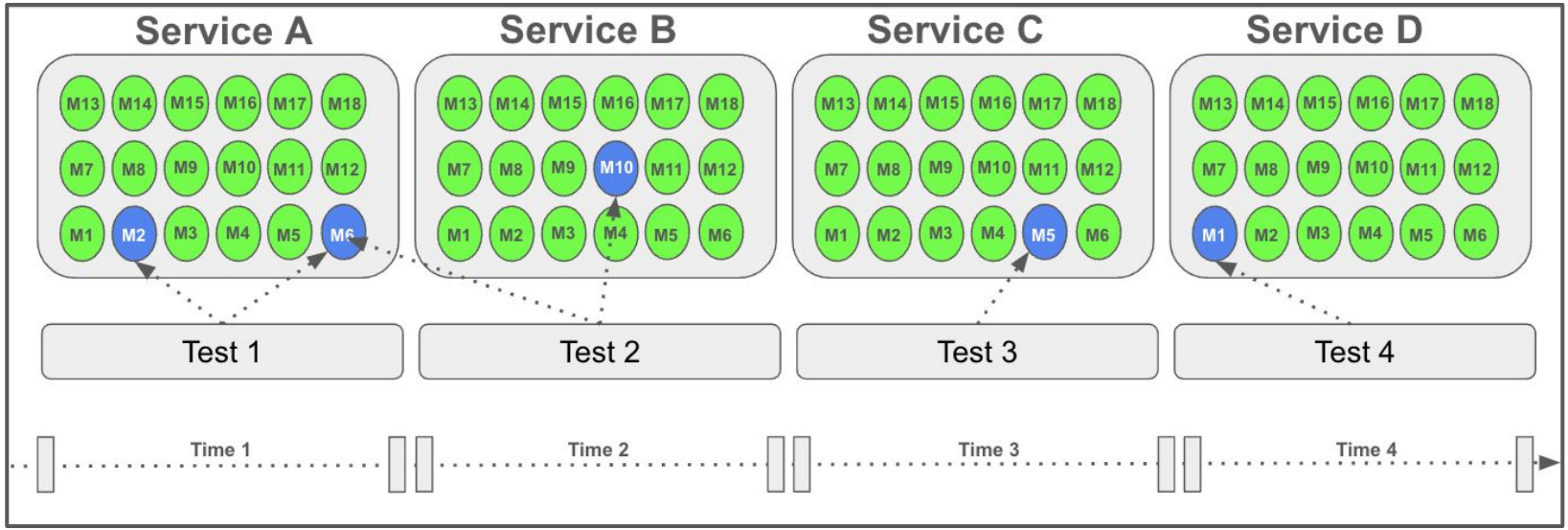
**What**:
Running tests on their own (in sequence) allows Sealights to improve the code to test mapping as only the methods hit by a test are triggered.
Marking these runs as calibrated give this information priority over non-calibrated runs
**Why:**
More precise correlation between code and tests
**Value**
Higher value as information from calibrated runs are much closer to the actual code to test mapping
**Trade Off:**
Test runs can be longer as tests are run in sequence and some modifications might need to be made to the test jobs to accommodate this

# Calibration Model

TTIA Training




Service A
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Service B
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Service C
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6

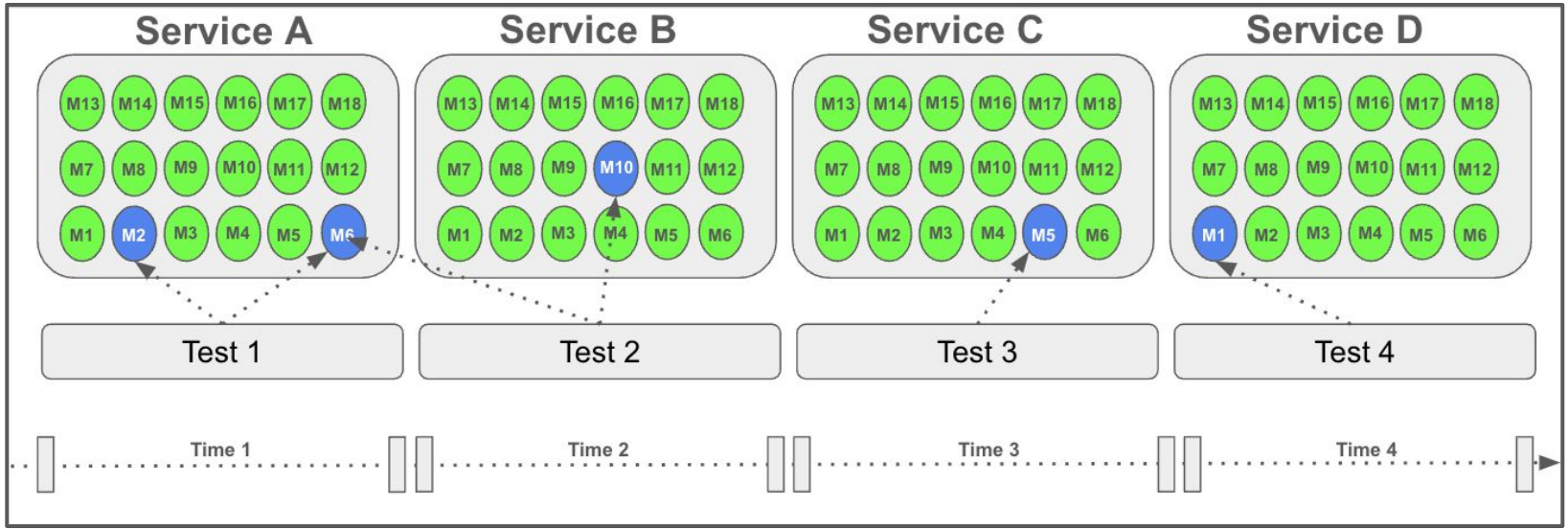



**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

# Scenario 1:

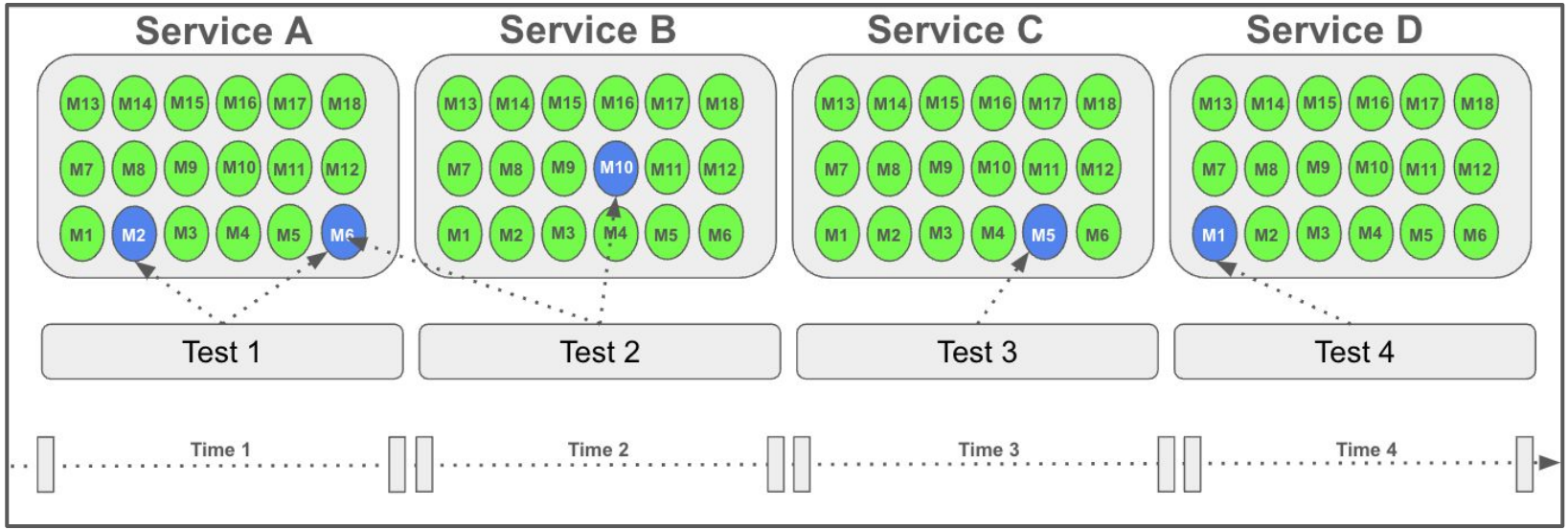
No Changes: 100% Savings




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 1:
## One method changed




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 1: No method changed

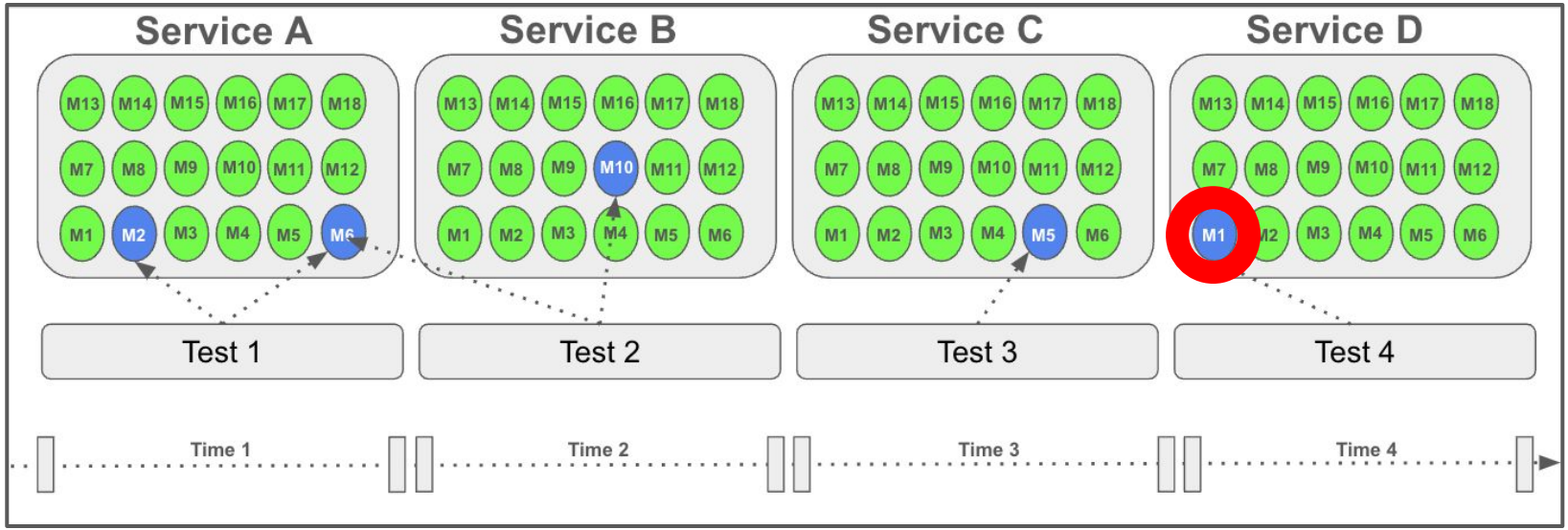
Recommendation List

## 100% Savings

| Test | Method |
|---|---|
| Test1 | Skip |
| Test2 | Skip |
| Test3 | Skip |
| Test4 | Skip |

# Scenario 2:

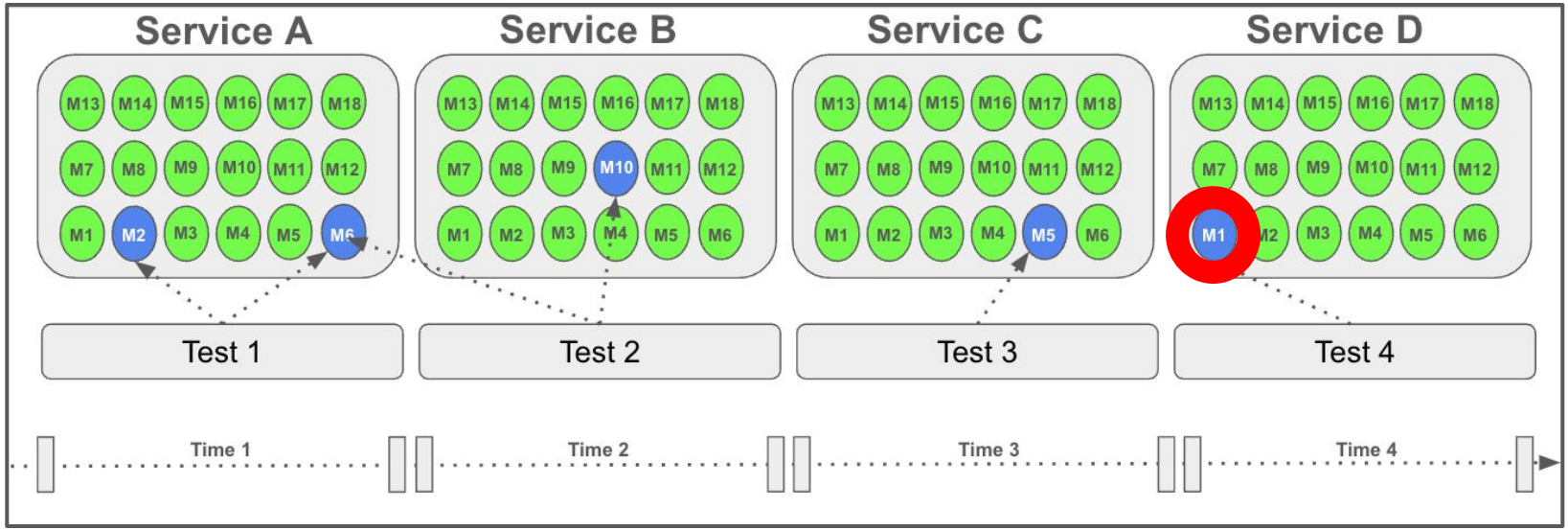
One Method Changed: 75% Savings




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| **D** | **M1** | **Test 4** |

## Scenario 2:
## One method changed




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 2: One method changed

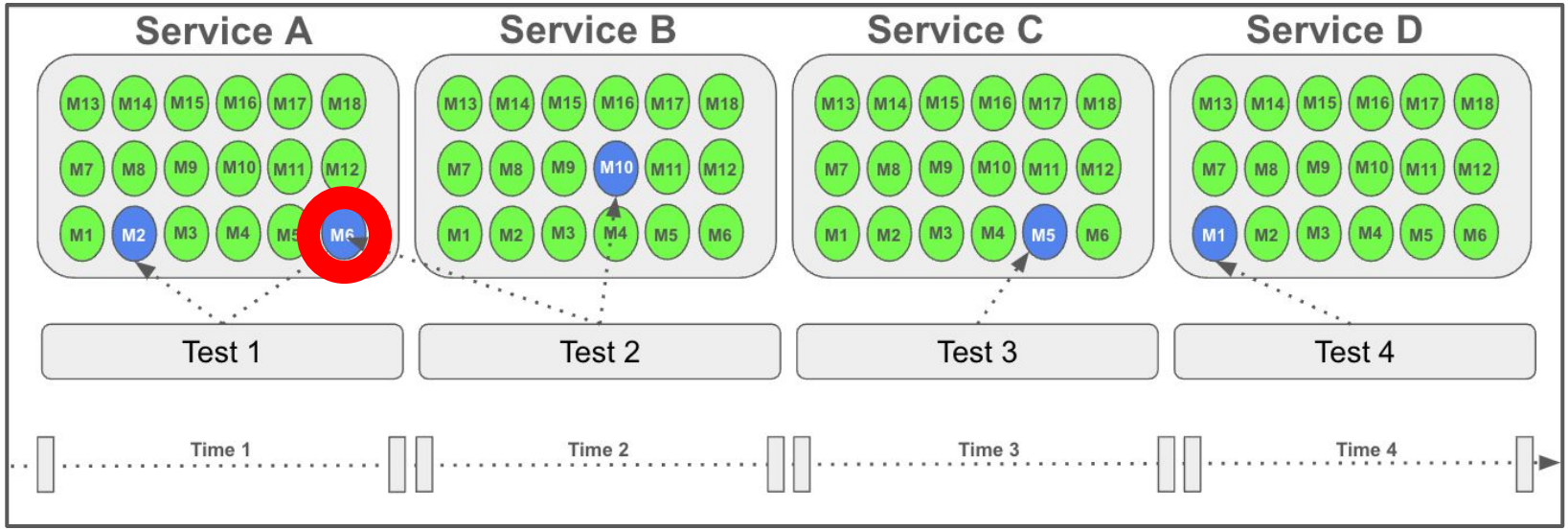
Recommendation List

**75% Savings**

| Test | Method |
|---|---|
| Test1 | Skip |
| Test2 | Skip |
| Test3 | Skip |
| Test4 | Execute |

# Scenario 3:

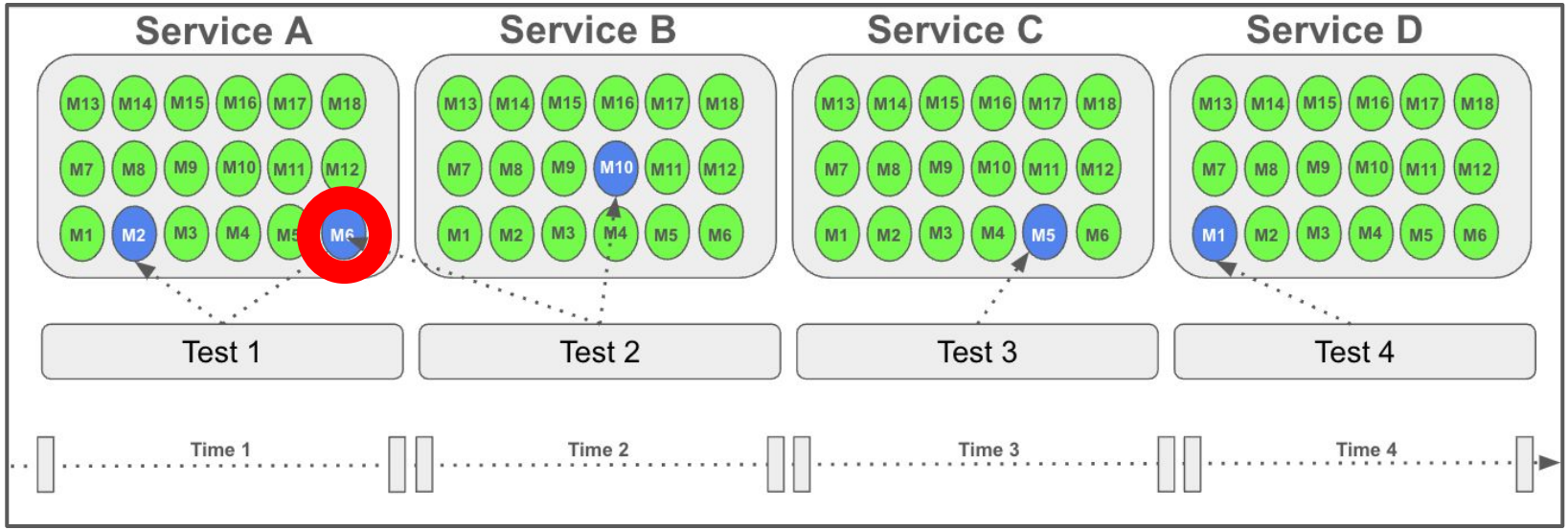
One Method Changed: 50% Savings




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 3:
## One method change




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 3: One method change

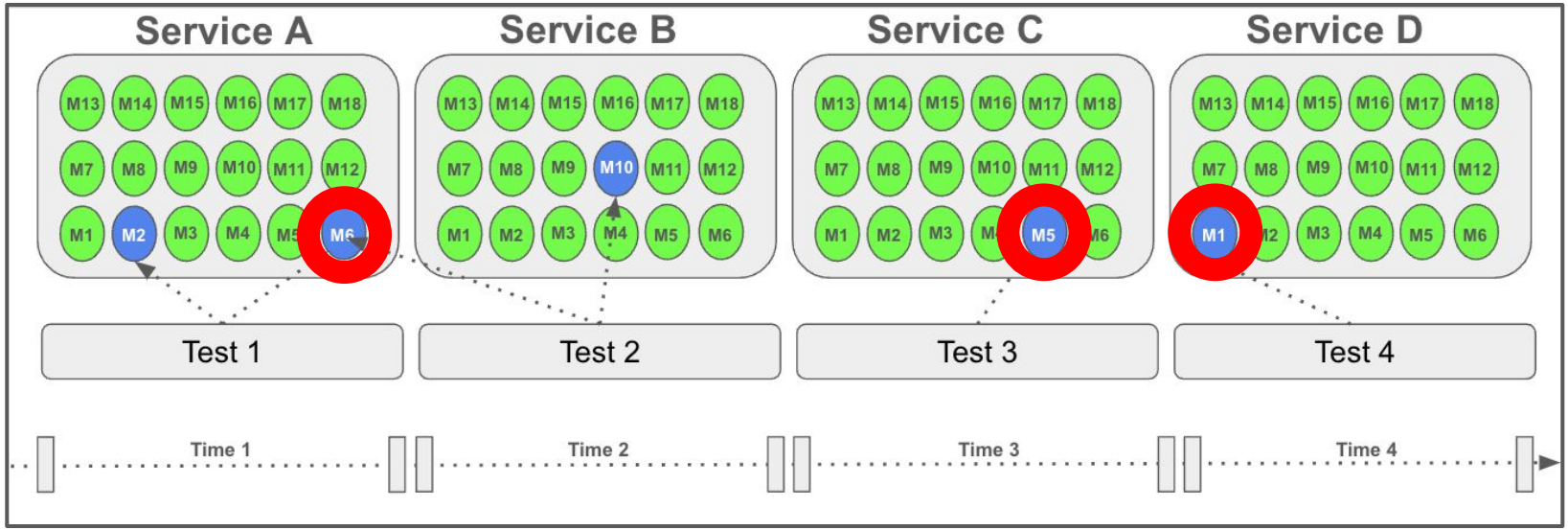
Recommendation List

**50% Savings**

| Test | Method |
|---|---|
| Test1 | Skip |
| Test2 | Execute |
| Test3 | Skip |
| Test4 | Execute |

# Scenario 4:

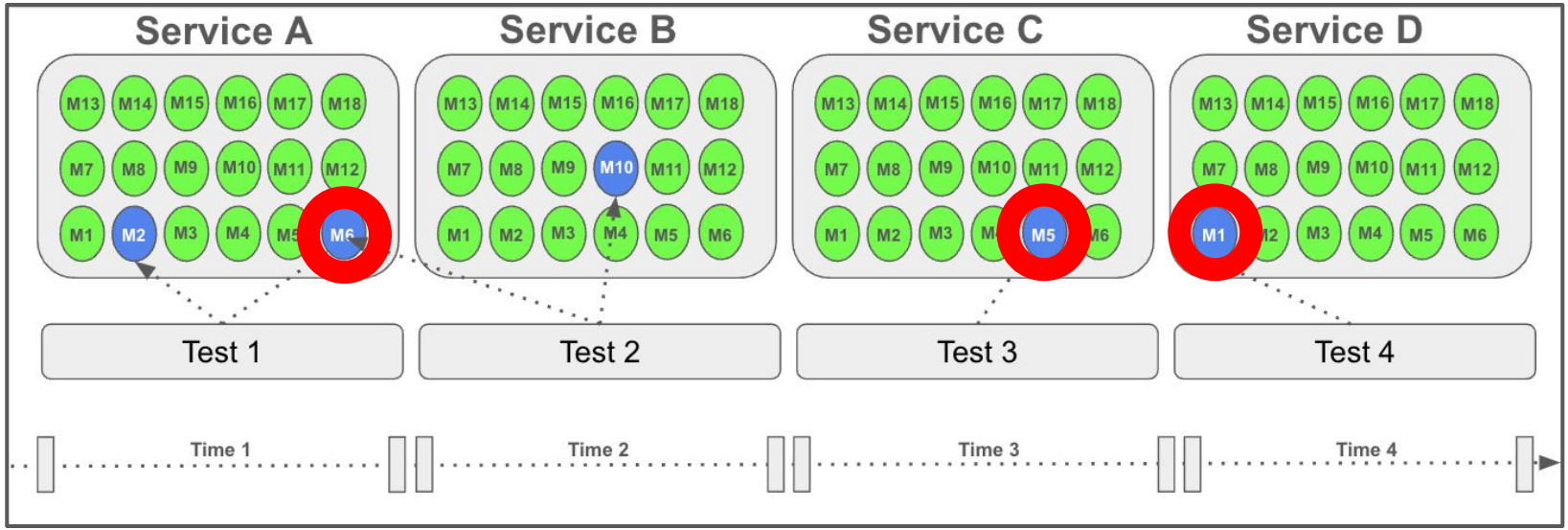
3 Methods Changed: 0% Savings




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 4:
## 3 methods changed




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 4: 3 methods changed

Recommendation List

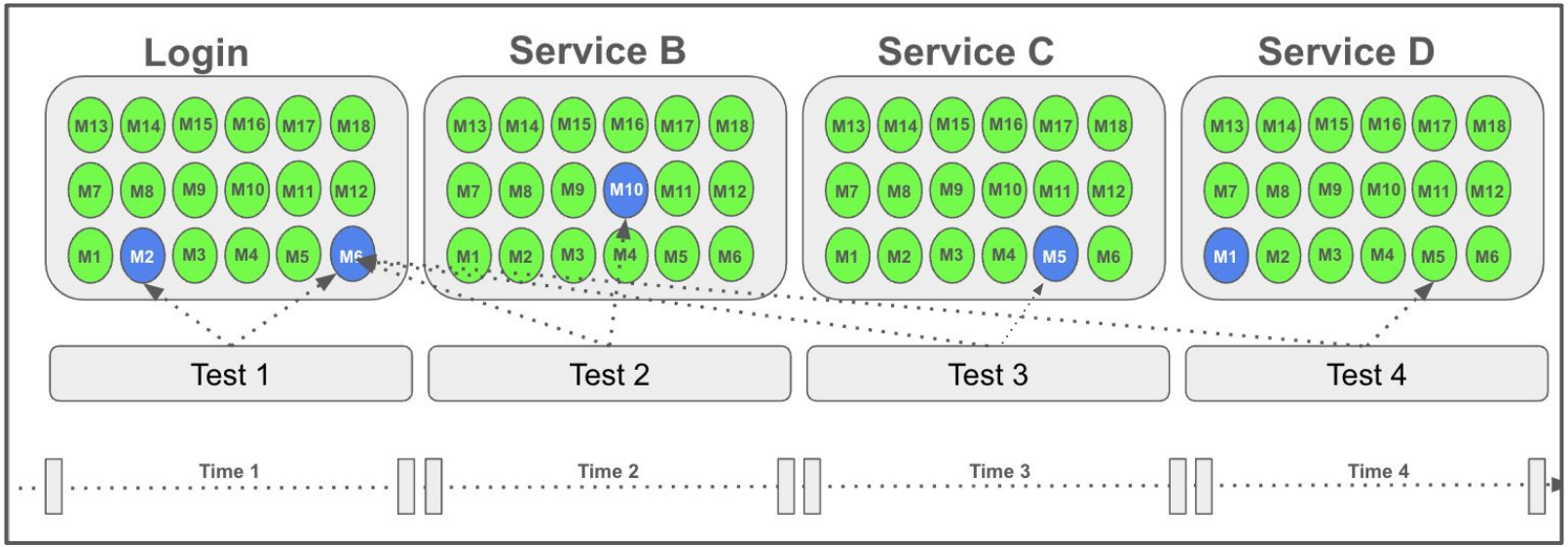
## 0% Savings

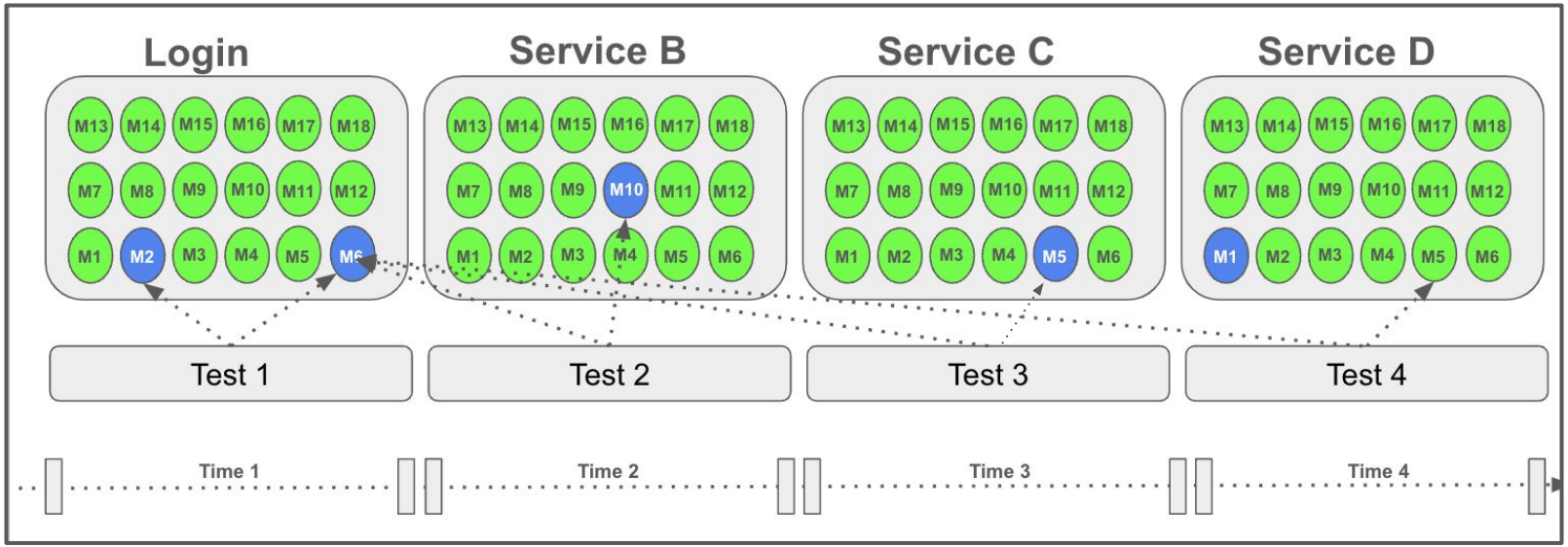
| Test | Method |
|---|---|
| Test1 | Execute |
| Test2 | Execute |
| Test3 | Execute |
| Test4 | Execute |

# Scenario 5 :

1 Methods Changed: 0% Savings

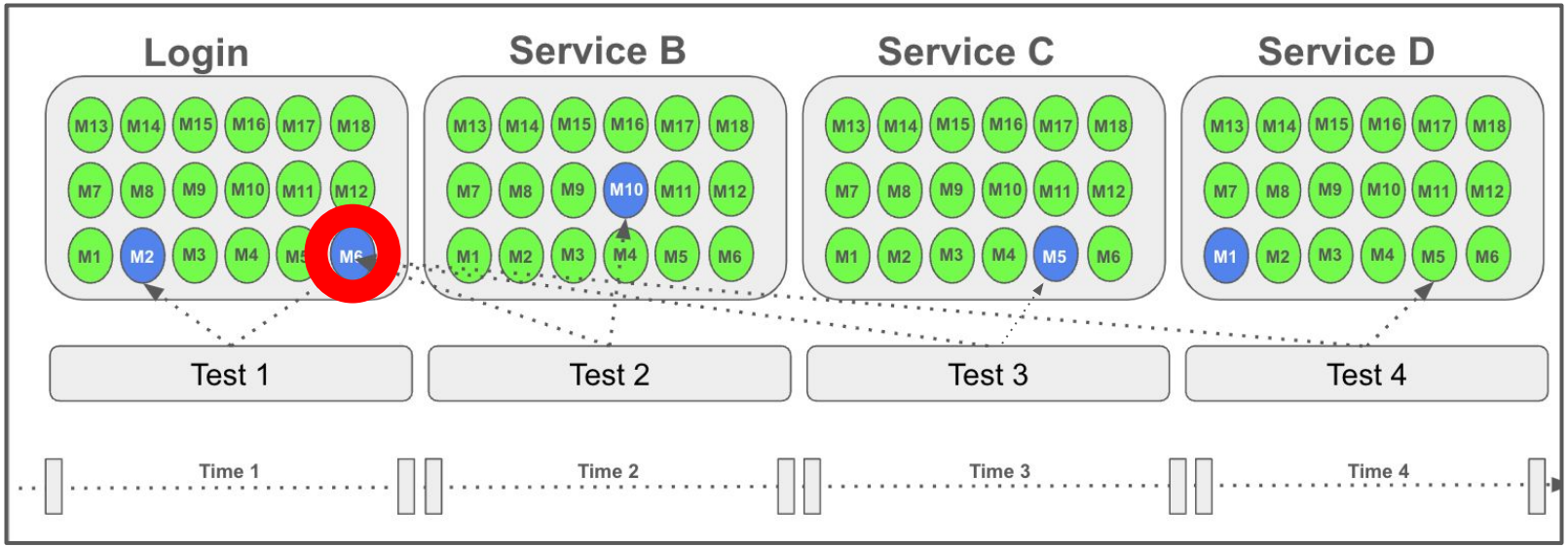
[Common Code Scenario]


Login
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4
Time 1
Time 2
Time 3
Time 4



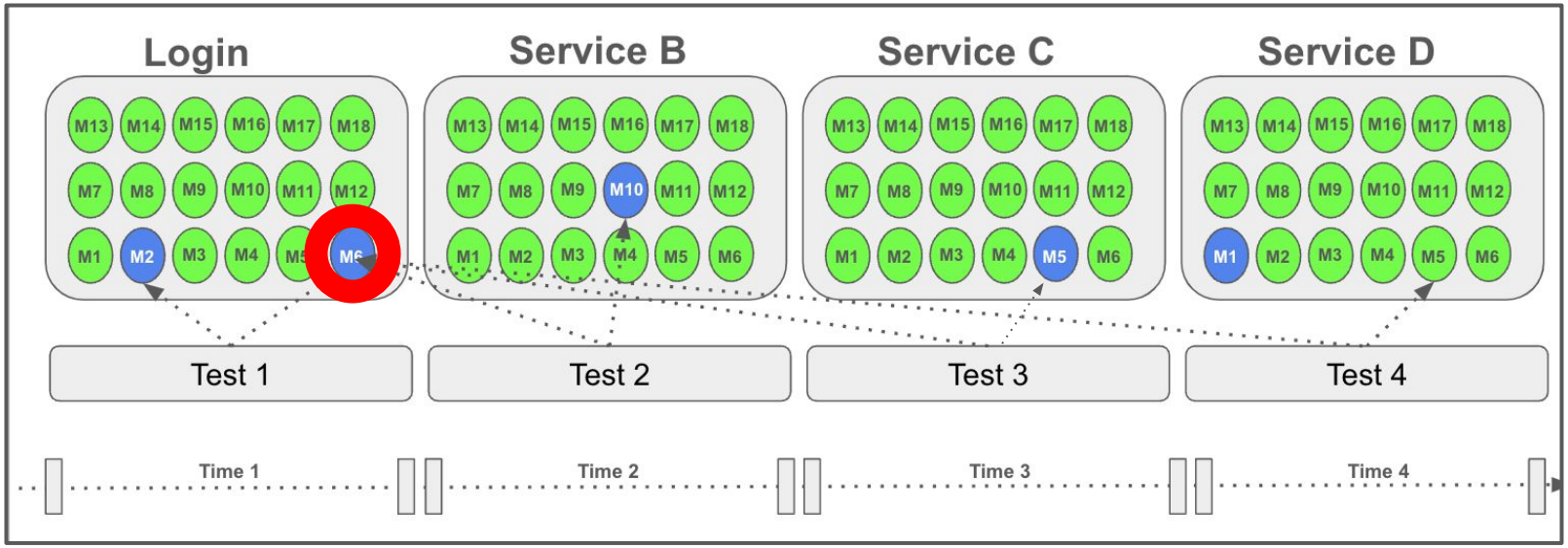

**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| Login | M2 | Test 1 |
| Login | M6 | Test 1, Test 2, Test 3, Test 4 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| Login | M2 | Test 1 |
| Login | M6 | Test 1, Test 2, Test 3, Test 4 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| Login | M2 | Test 1 |
| Login | M6 | Test 1, Test 2, Test 3, Test 4 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 5: 1 method changed

Recommendation List

## 0% Savings

| Test | Method |
|---|---|
| Test1 | Execute |
| Test2 | Execute |
| Test3 | Execute |
| Test4 | Execute |

# Configuration Option: Mark Common Code

Sealights

SeaLights provides the ability to mark methods as common code.

This can either be marked to be ignored, or to link specific tests to the common code.

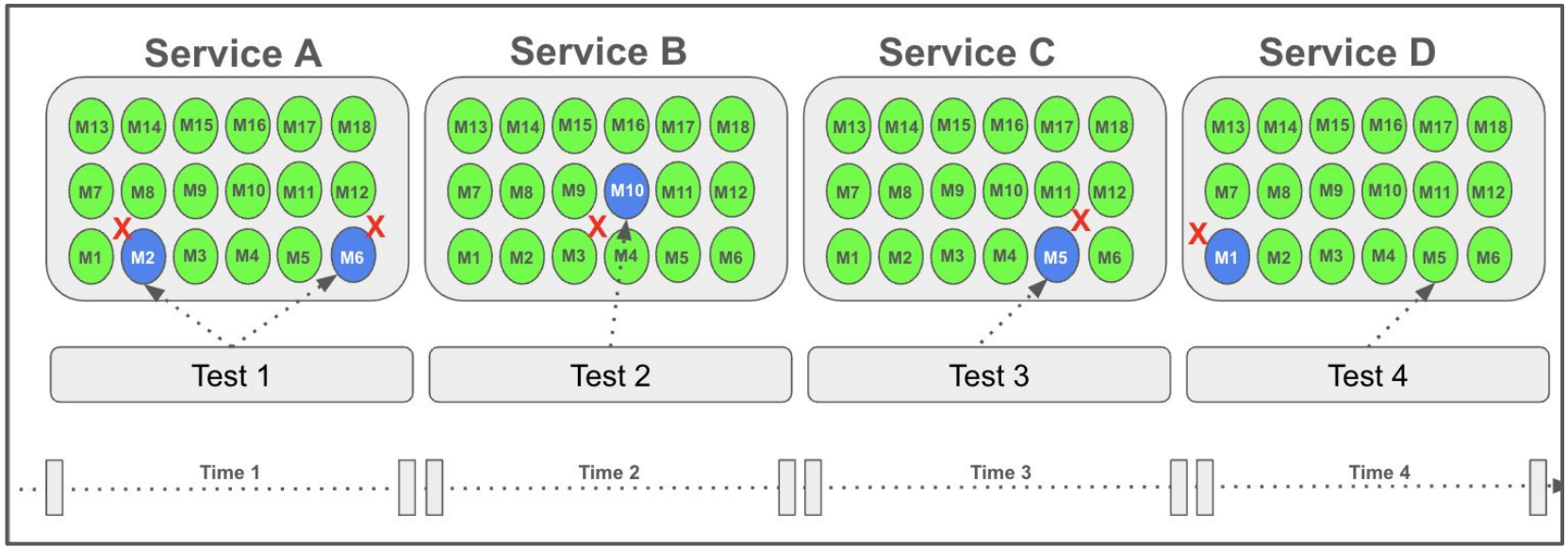
Once this is done, if any of the methods of the common code are modified, then they will either be completely ignored, or trigger a minimal set of tests instead of triggering almost all of them.

This can greatly improve the savings if the customers code contains a lot of common code.

# Scenario 6:

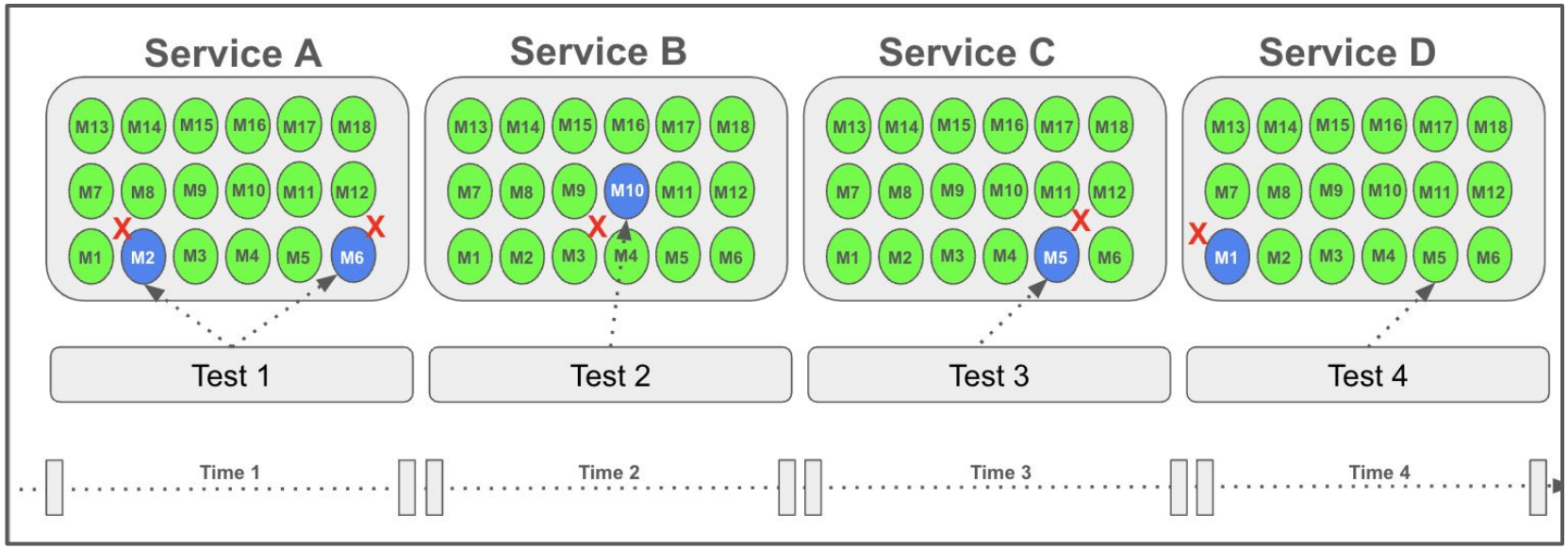
No Methods Changed: 0% Savings

[Fail Test Scenario]




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |

## Scenario 6: No method changed

Recommendation List

## 0% Savings

| Test | Method |
|---|---|
| Test1 | Execute |
| Test2 | Execute |
| Test3 | Execute |
| Test4 | Execute |

# Configuration Option: Ignore Fail Test

Sealights

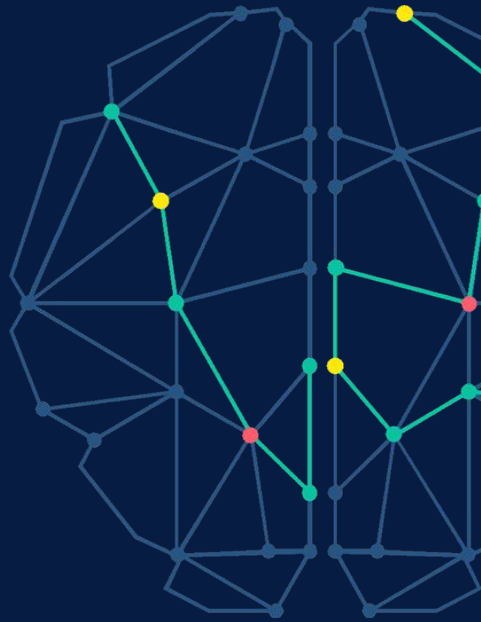
SeaLights by default not recommend a test that failed to be skipped until it has passed.

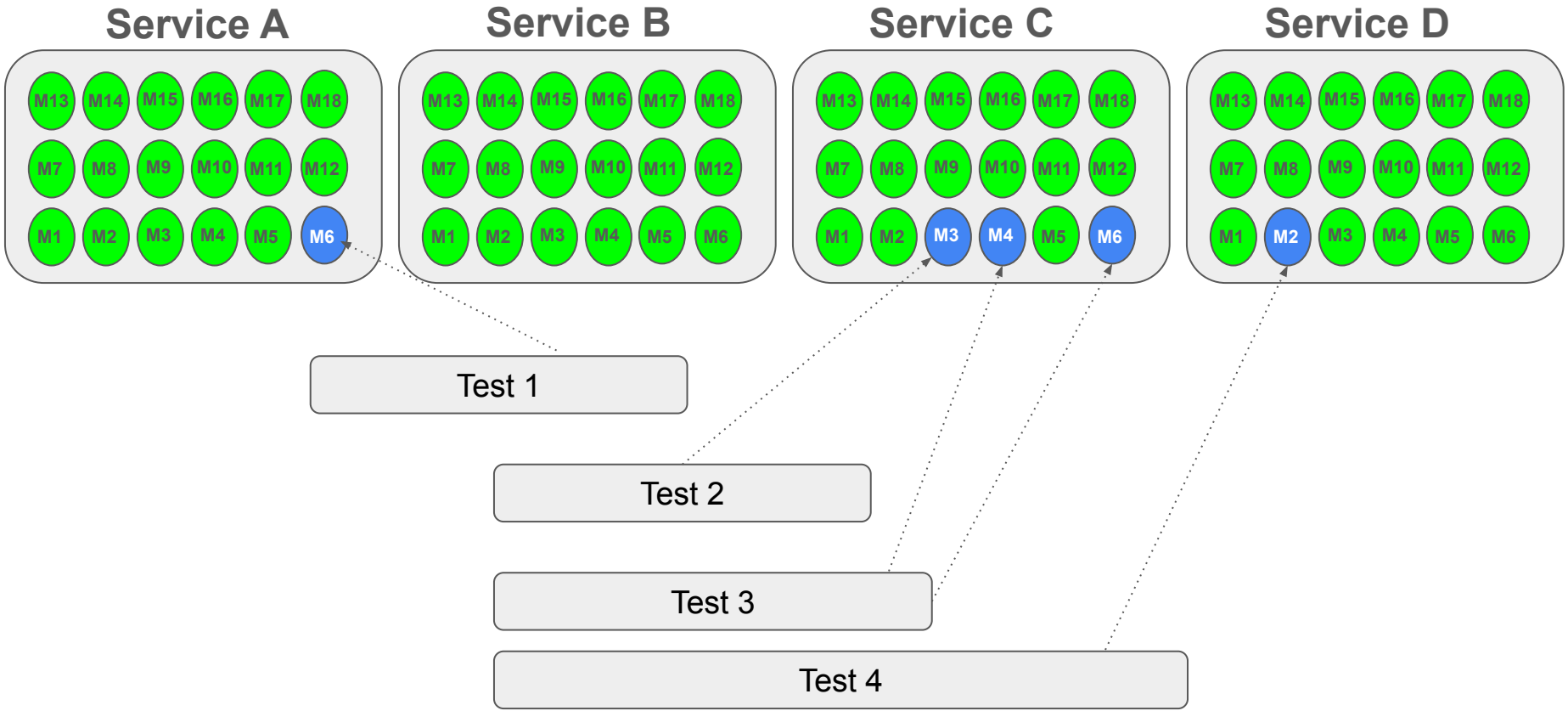
This is due to the fact that if the failure was caused by a real issue, then it must pass before it can be skipped.

As in some environments, failed tests could be common and less important to customers, Sealights provides the ability to update the policy and ignore if a test failed or not and rely only on the code to test mapping and modified methods.

# Statistical Model

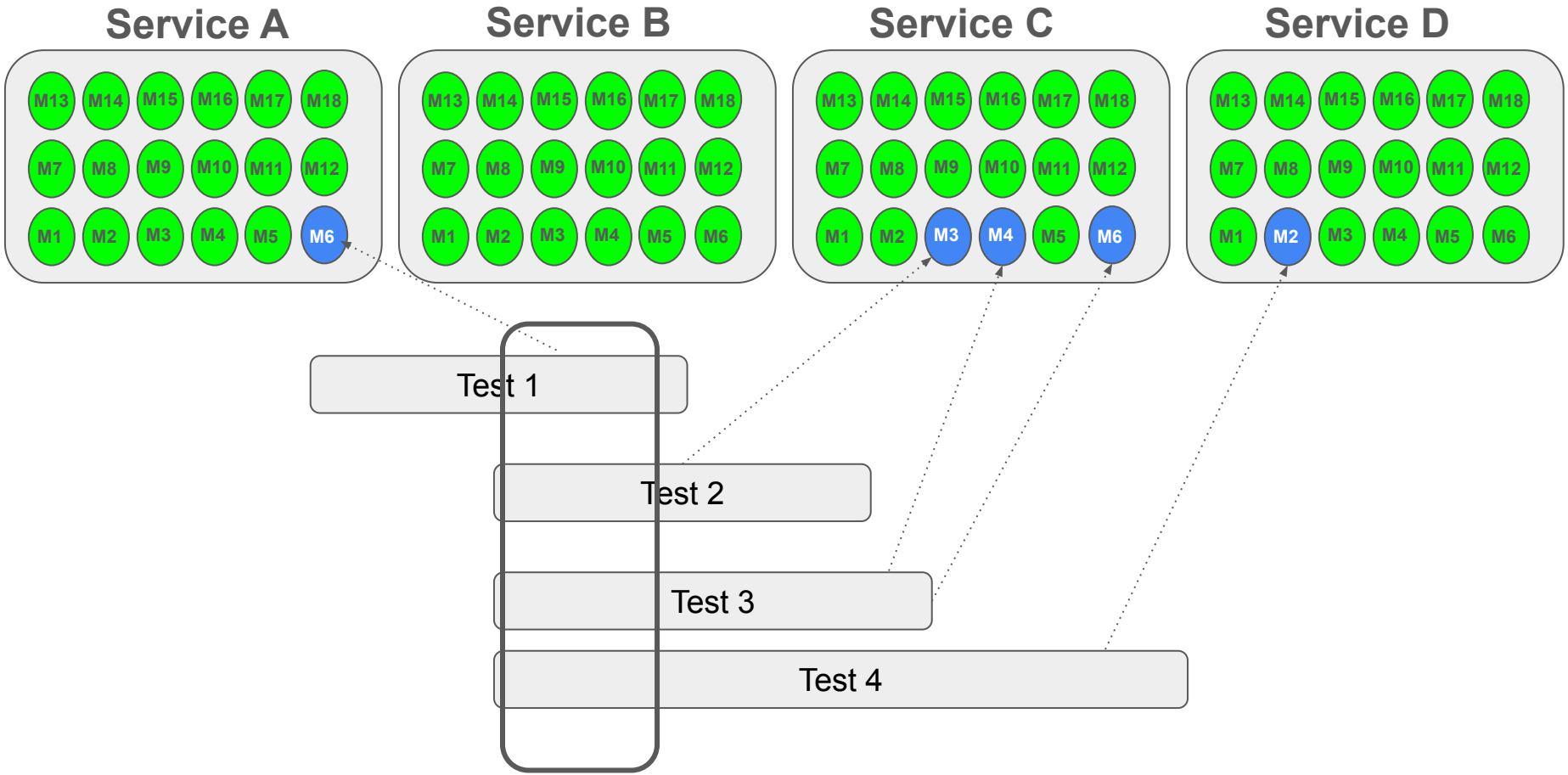
TIA Training

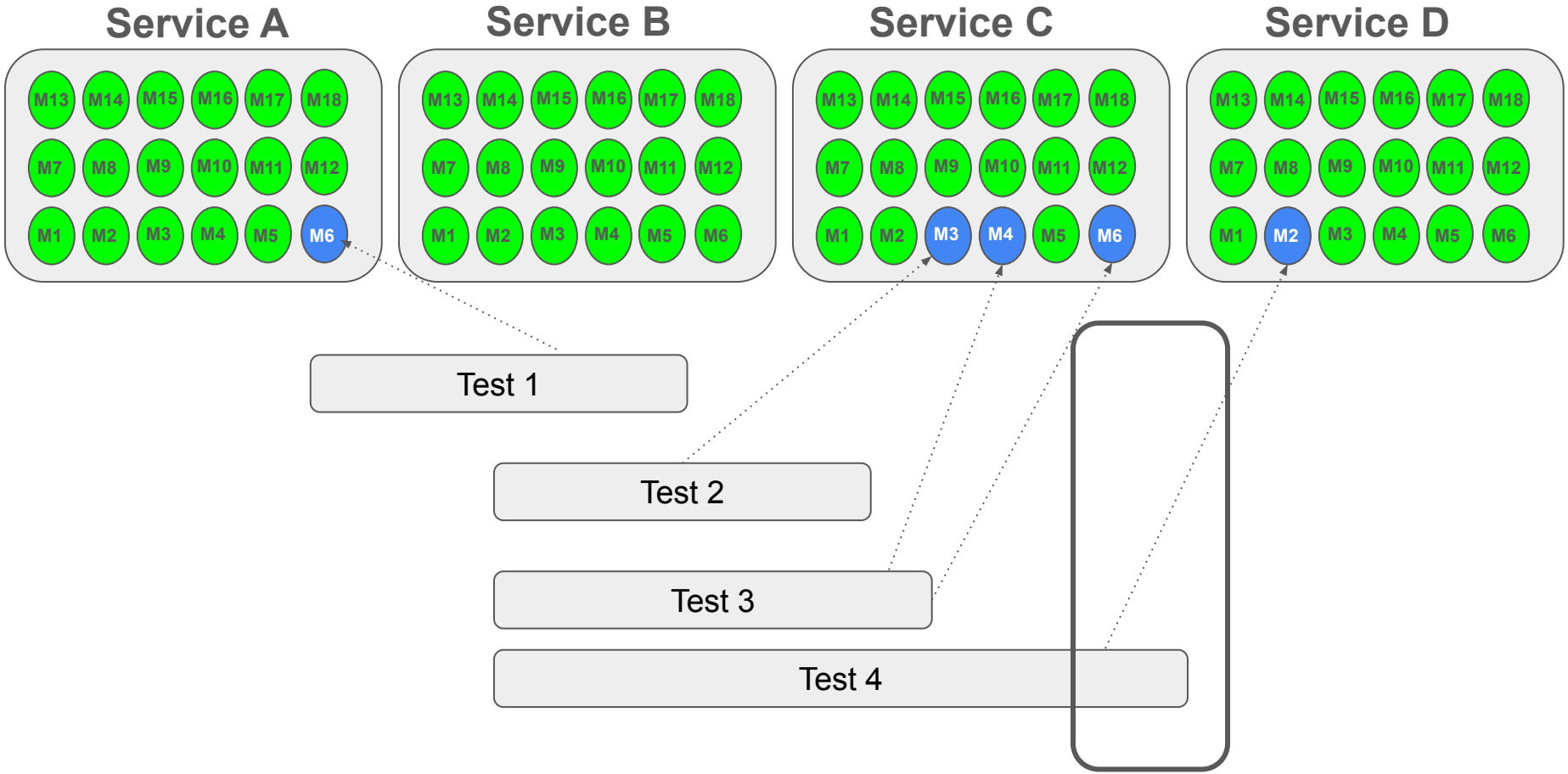


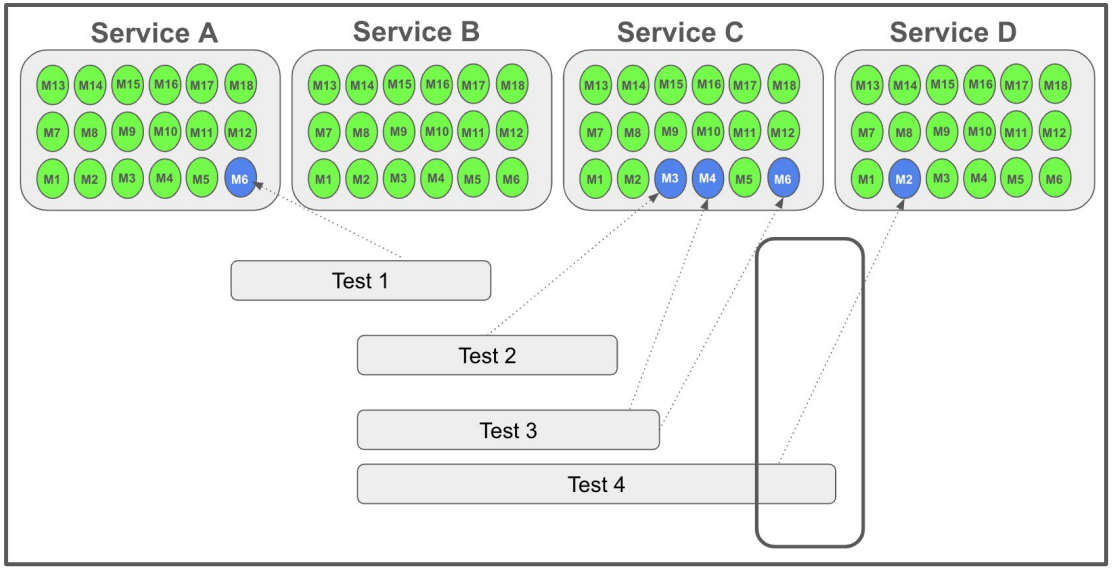

Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4

# Scenario 1:

1 Method change: 75% Savings

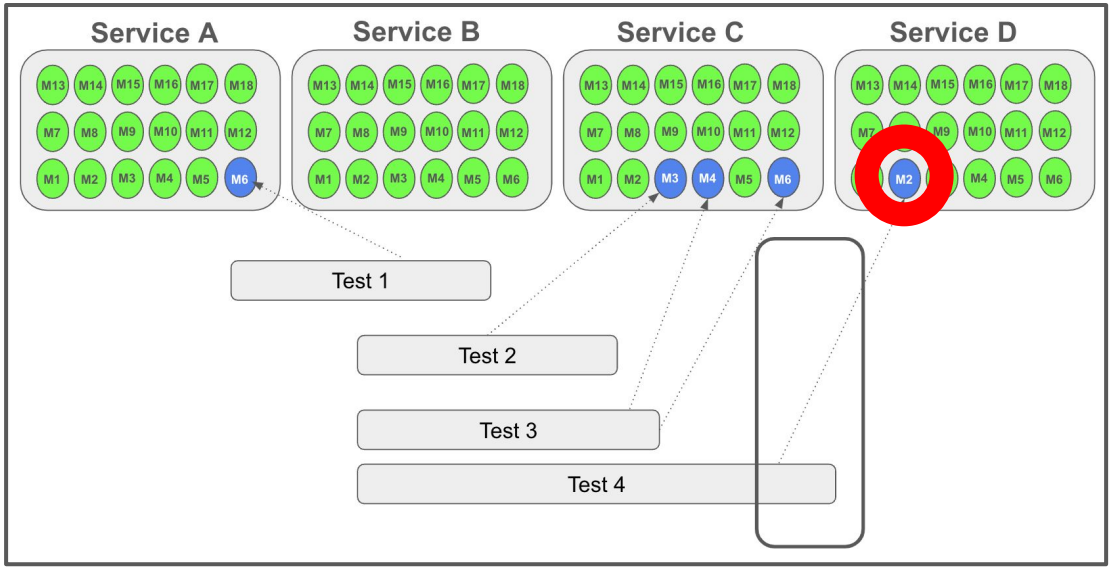

Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4


Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| C | M3 | Test 2 |
| C | M4 | Test 3 |
| C | M6 | Test 3 |
| D | M2 | Test 4 |




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| C | M3 | Test 2 |
| C | M4 | Test 3 |
| C | M6 | Test 3 |
| D | M2 | Test 4 |

## Scenario 1: 1 method changed

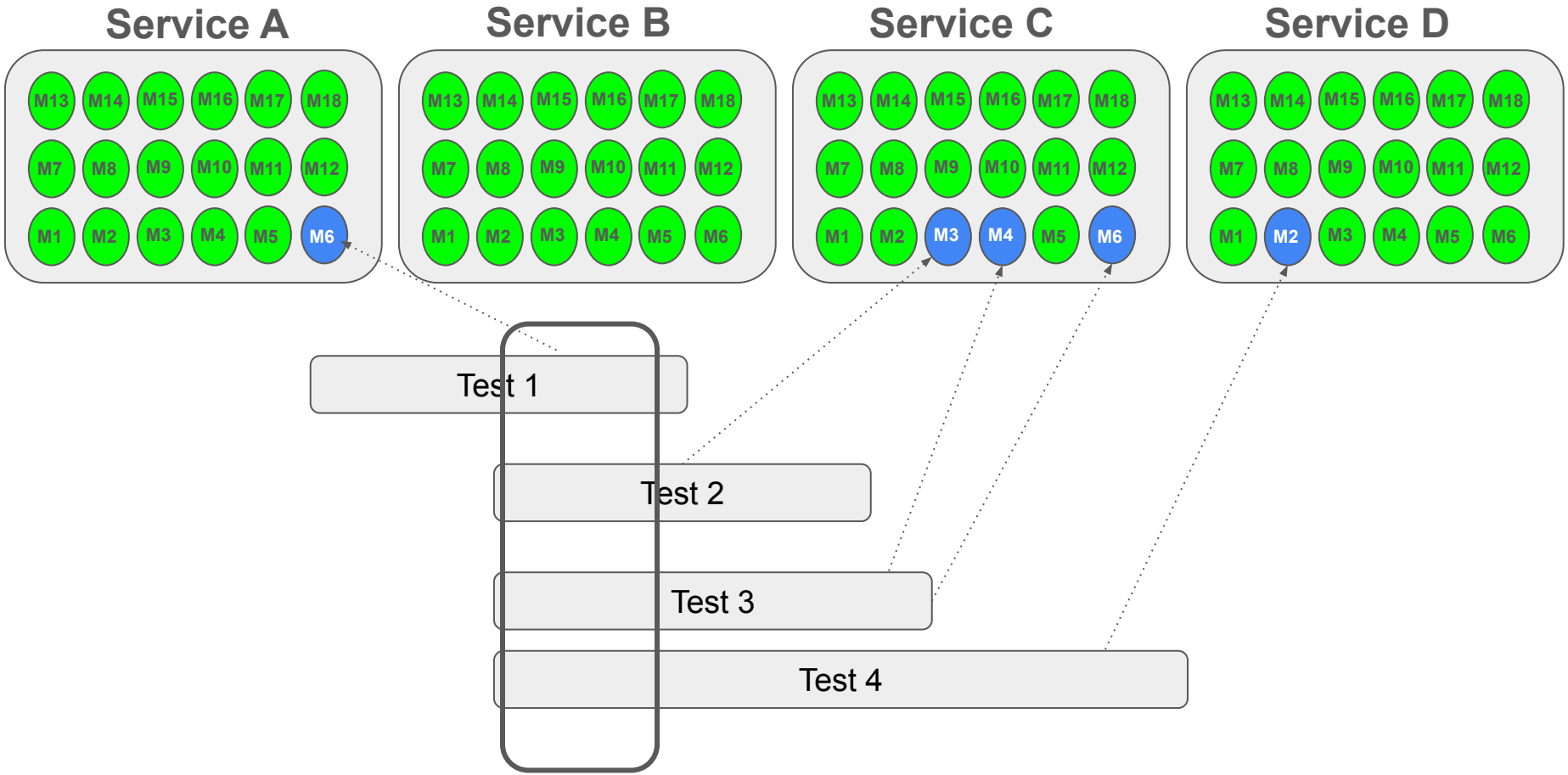
Recommendation List

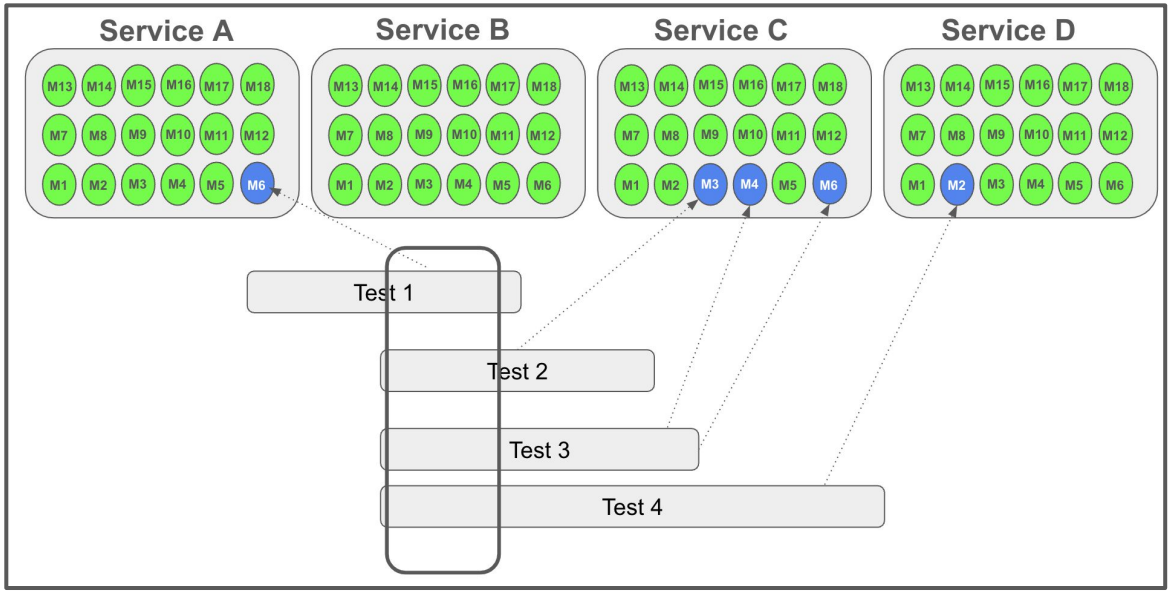
## 75% Savings

| Test | Method |
|---|---|
| Test1 | Skip |
| Test2 | Skip |
| Test3 | Skip |
| Test4 | Execute |

# Scenario 2:

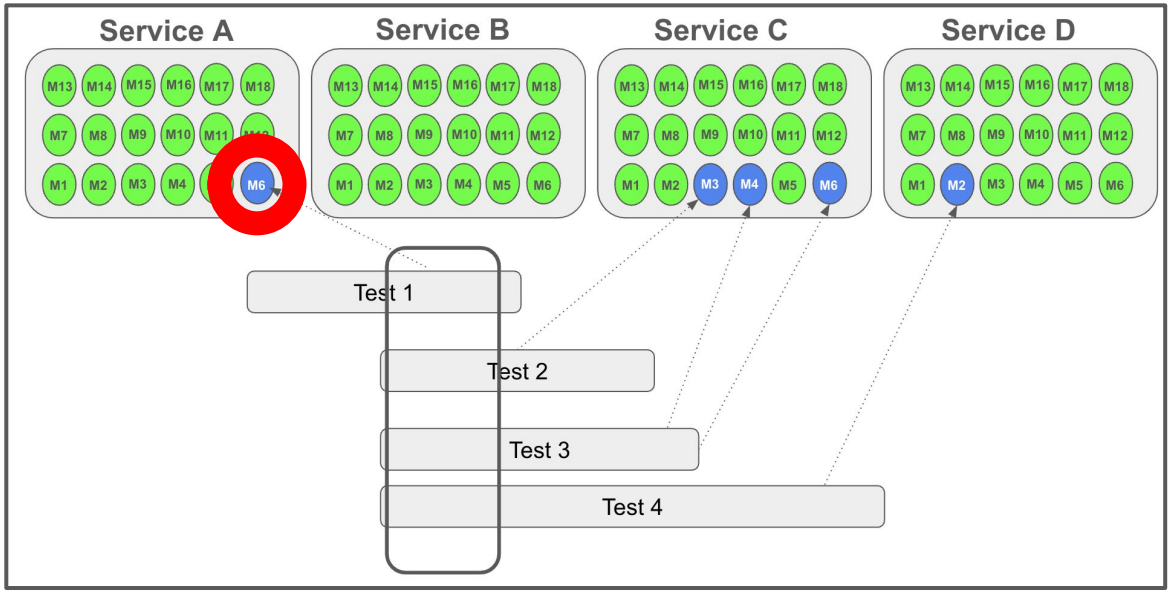
1 Method change: 0% Savings


Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| C | M3 | Test 2 |
| C | M4 | Test 3 |
| C | M6 | Test 3 |
| D | M2 | Test 4 |




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| C | M3 | Test 2 |
| C | M4 | Test 3 |
| C | M6 | Test 3 |
| D | M2 | Test 4 |

## Scenario 2: 1 method changed

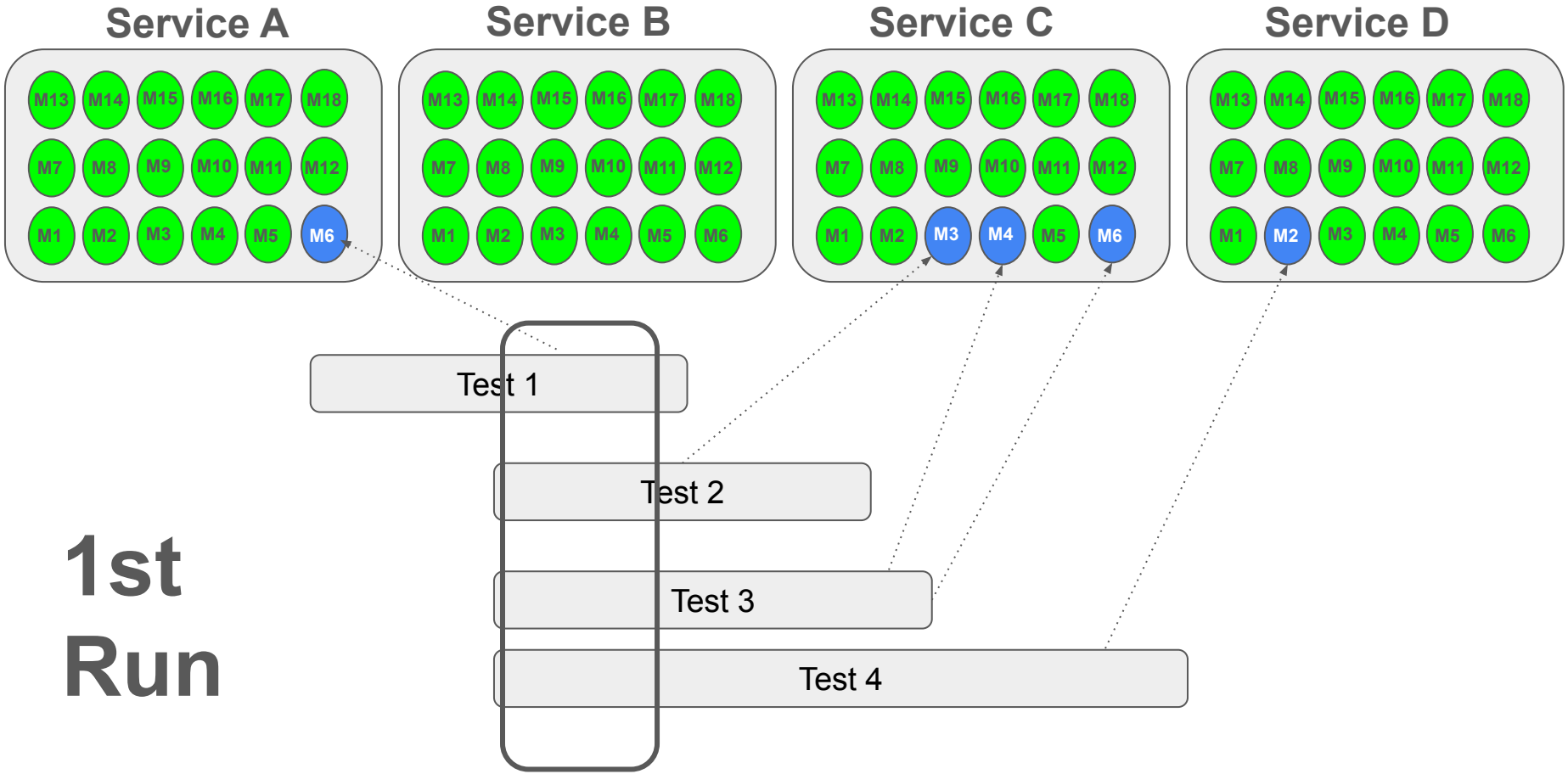
Recommendation List

## 0% Savings

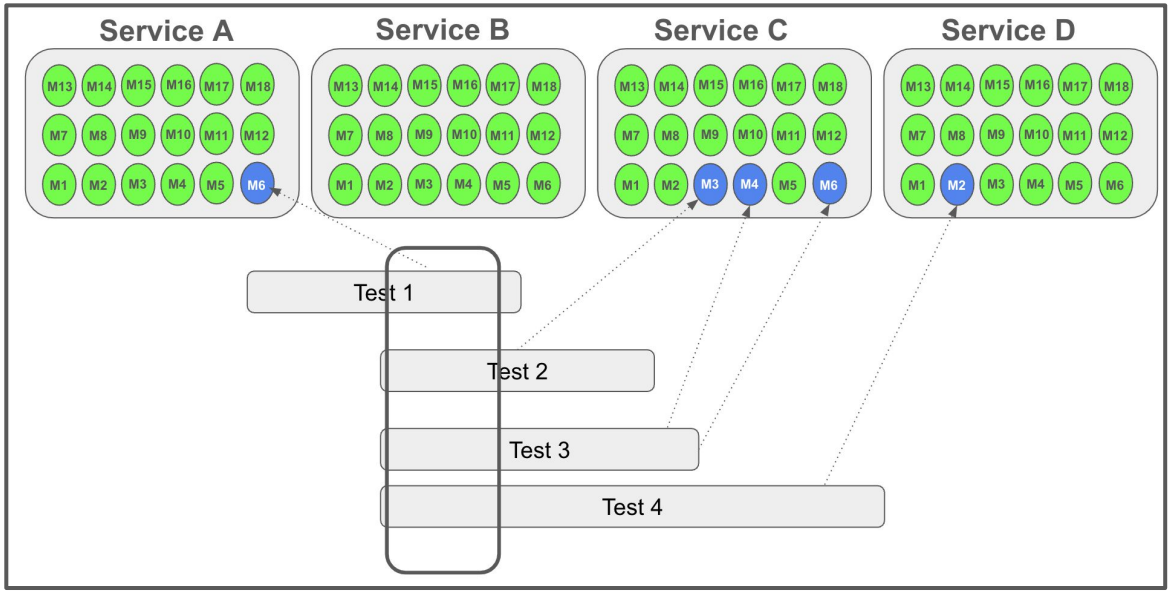
| Test | Method |
|---|---|
| Test1 | Execute |
| Test2 | Execute |
| Test3 | Execute |
| Test4 | Execute |

# Scenario 3:

Improve statistical model over time

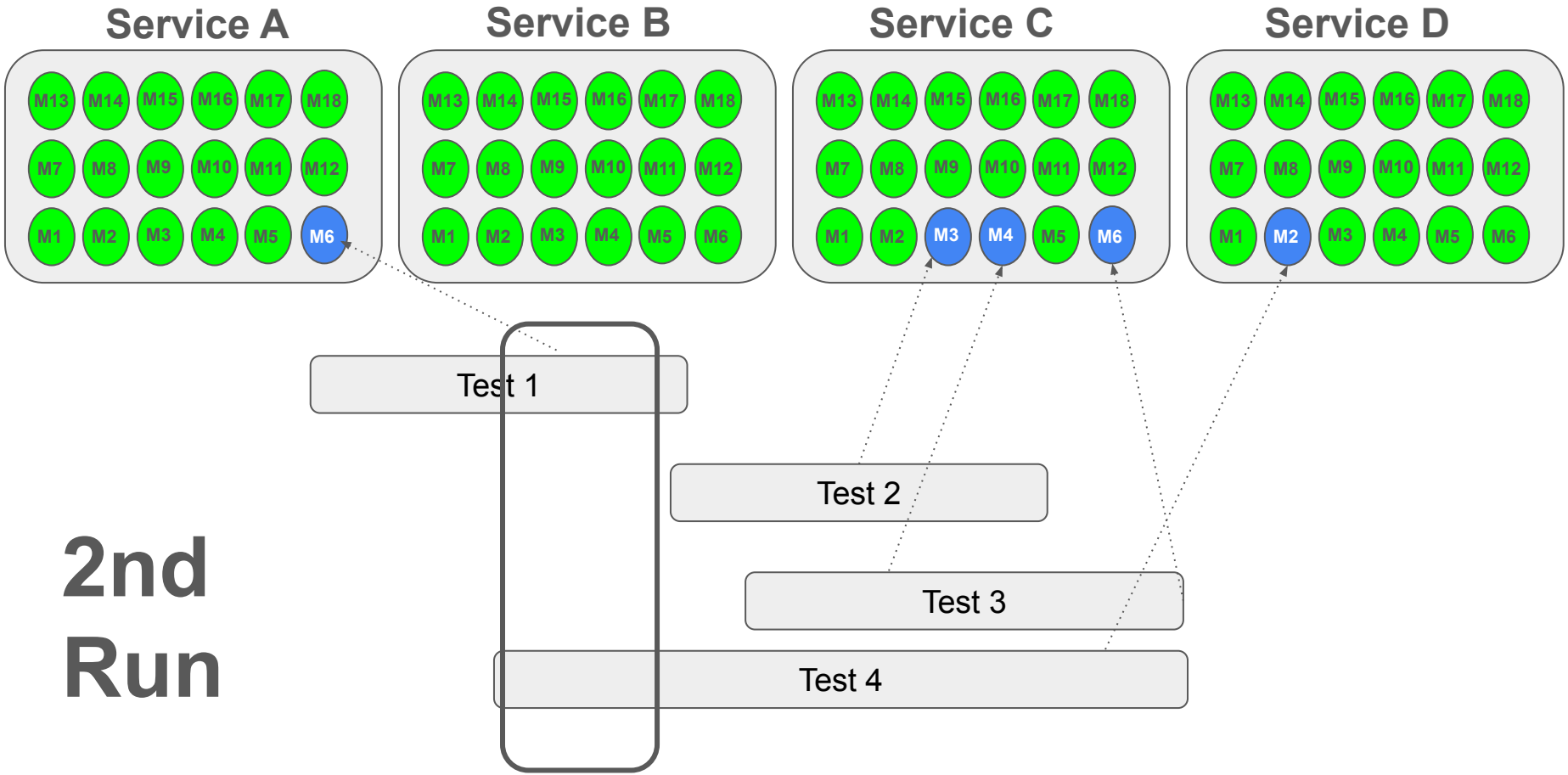

Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4
1st
Run

# 1st Run

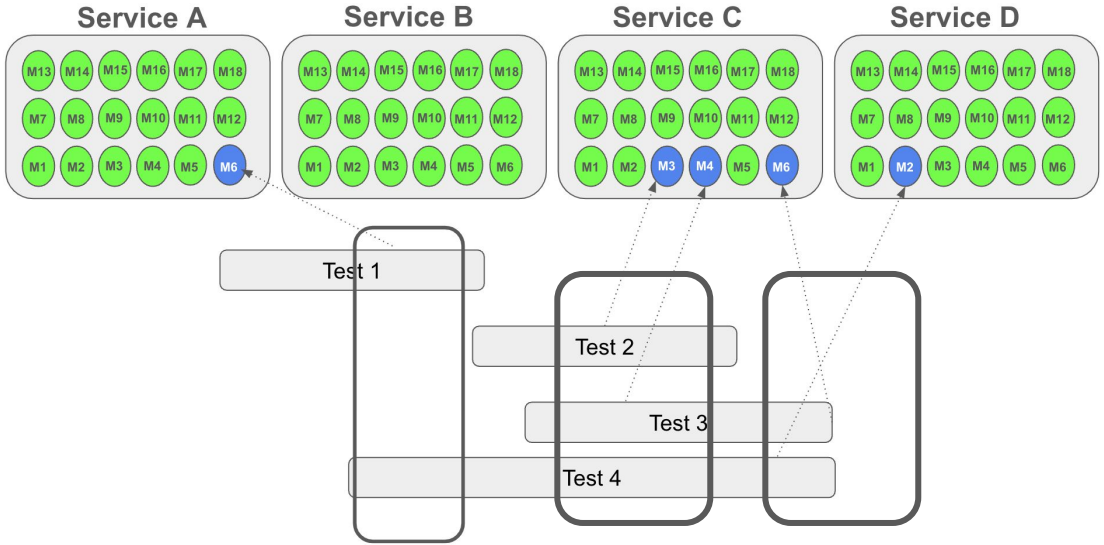



**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| C | M3 | Test 2 |
| C | M4 | Test 3 |
| C | M6 | Test 3 |
| D | M2 | Test 4 |


Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4
2nd Run

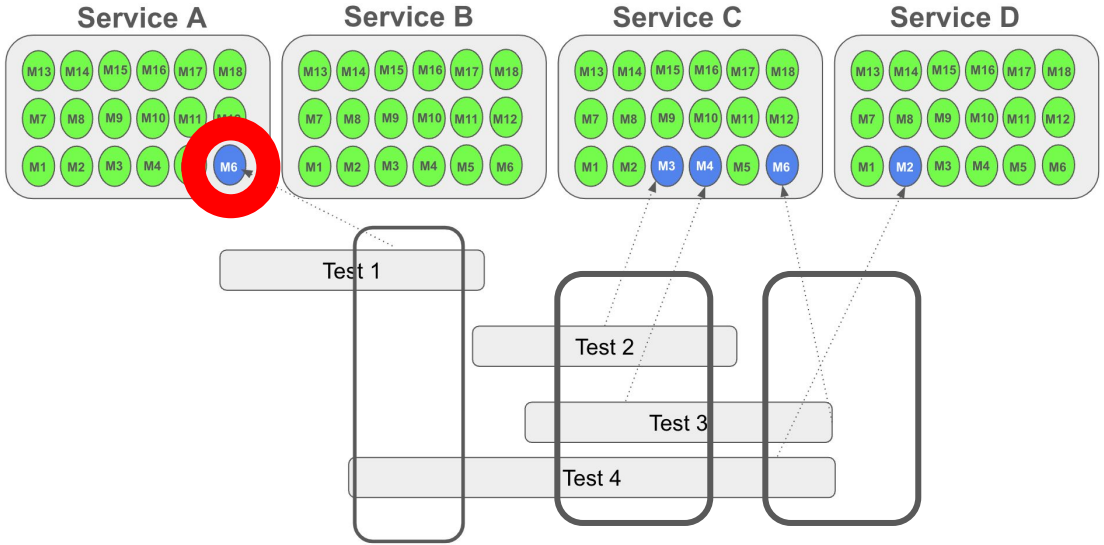
# 2nd Run




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 4 |
| C | M3 | Test 2, Test 3, Test 4 |
| C | M4 | Test 2, Test 3, Test 4 |
| C | M6 | Test 3, Test 4 |
| D | M2 | Test 3, Test 4 |

## 2nd Run




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 4 |
| C | M3 | Test 2, Test 3, Test 4 |
| C | M4 | Test 2, Test 3, Test 4 |
| C | M6 | Test 3, Test 4 |
| D | M2 | Test 3, Test 4 |

## Scenario 3: 1 method changed

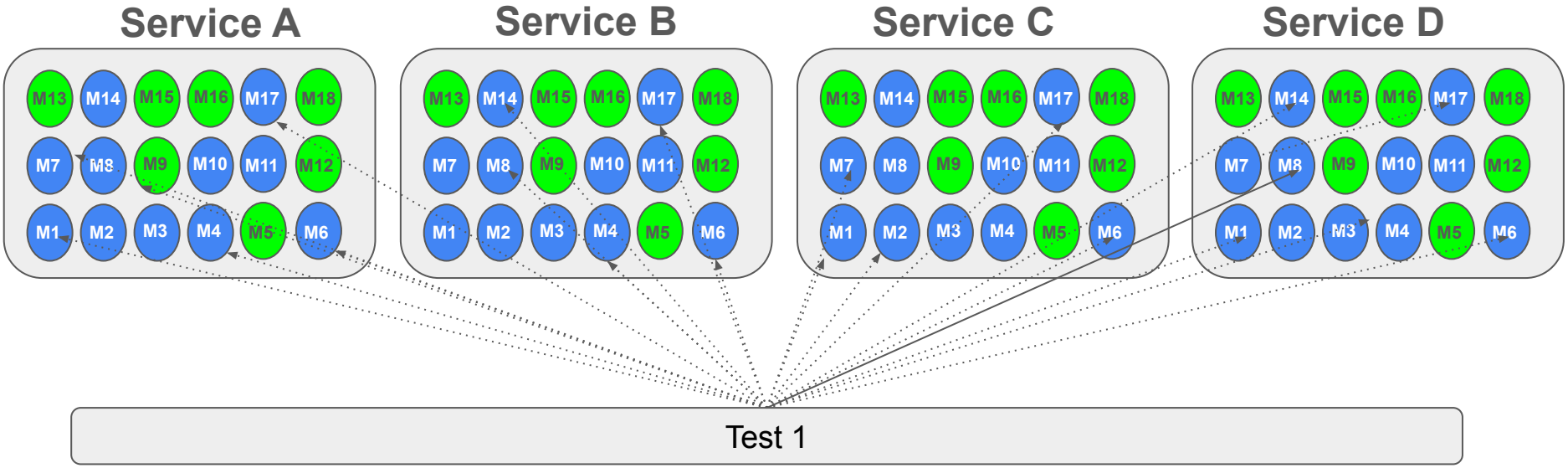
Recommendation List

## 50% Savings

| Test | Method |
|---|---|
| Test1 | Execute |
| Test2 | Skip |
| Test3 | Skip |
| Test4 | Execute |

# Scenario 4:

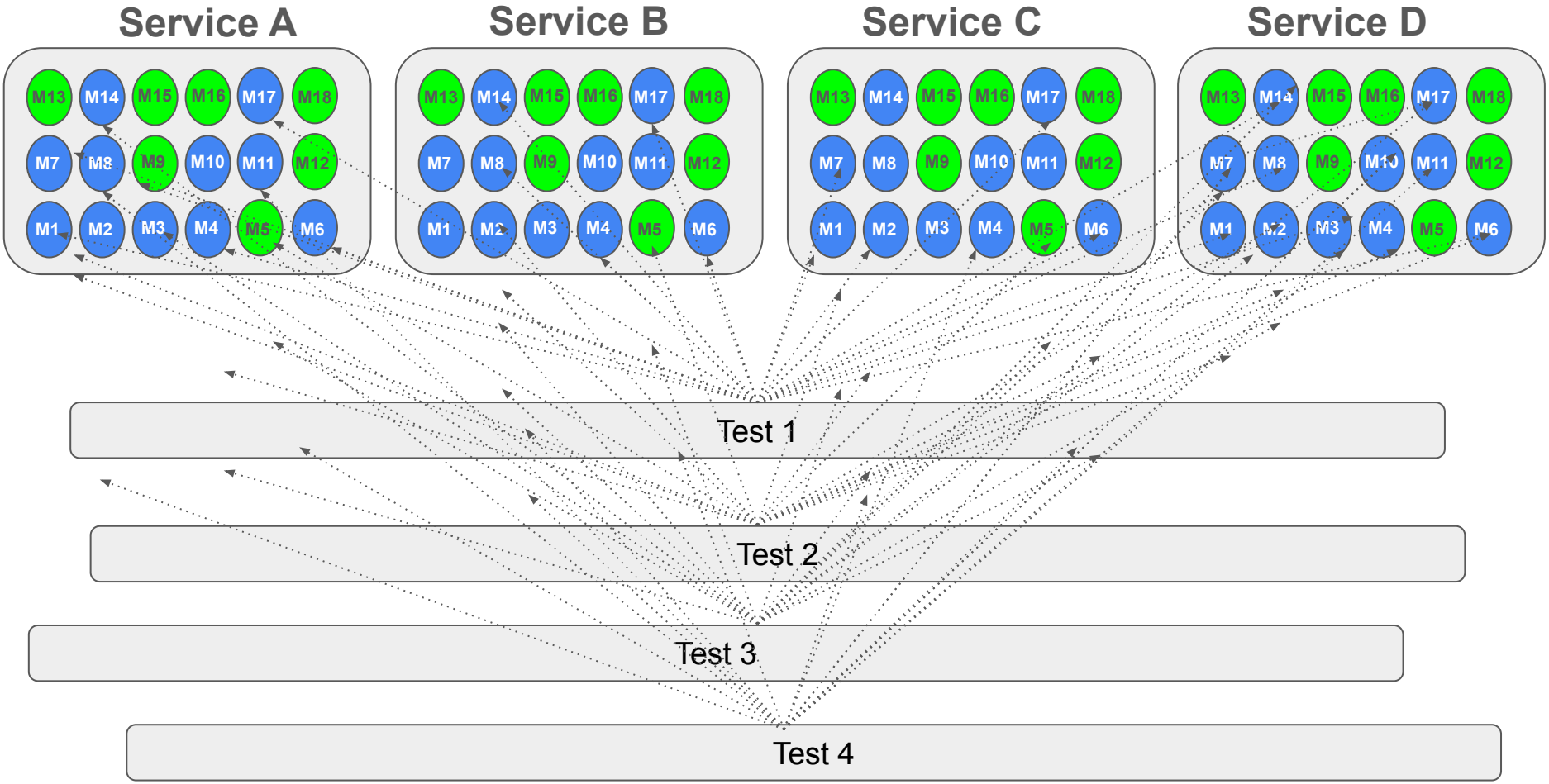
Long Parallel tests




Long test will trigger in most cases when there is a significant build changes

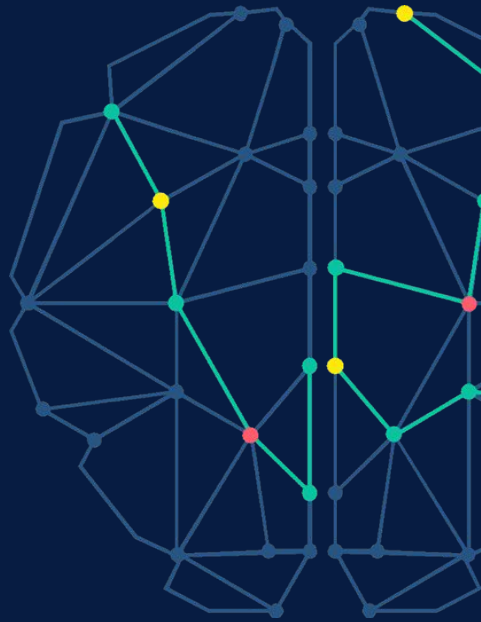
# Scenario 5:

The challenge of parallel execution and long tests


Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4

# Best Practice

TIA Training



# Best Practice

Sealights

The % savings of the TIA recommendation agent depends on the following parameters:

1. How much the test suite running in parallel vs in sequence
2. % of Common code
3. Length of the tests
4. % of failed Tests
5. Number of code changes per build

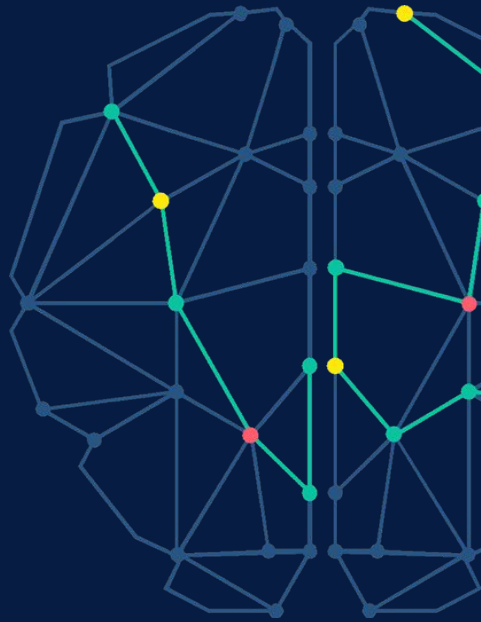
# Best Practice

Sealights

1. It is recommended to run without any changes to the testing cycles and processes, for a few weeks, to understand the quality of the Test to code mapping model minimum impact on the users - we call this period tuning period
2. After few weeks, based on the quality of the Test to code mapping and the statistical model, we will start to tune the system:
   a. Common code configuration
   b. Fail Test Configuration
3. If the improvement through statistical model doesn't improve the savings or further improvements are required, then the next option is to start calibration executions
   a. Periodically run the Test suites in sequence and lock the model by reporting them as a calibration run to Sealights, this way the test to code mapping is 100% accurate
   b. If the calibration works well and improve the savings, this needs to continue to be provided periodically to allow SeaLights to learn new code mappings and new tests
4. Between any calibration run you can continue working using the statistical model. SeaLights will combine the best score per method before recommending tests
5. In Case of very long running e-2-e tests, it is recommended to break them into more shorter tests if possible. This can further improve the Test to code mapping model
6. When tests are always run in parallel at the same time, the statistical model will not improve. Randomizing the start time or order of the tests can greatly improve the statistical model

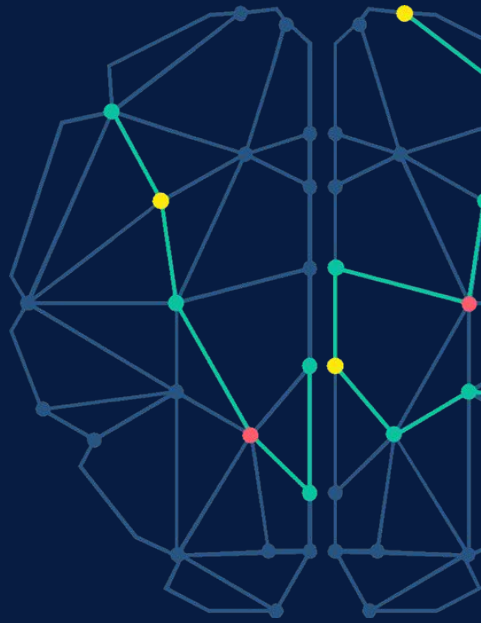
# Thank You

TIA Training



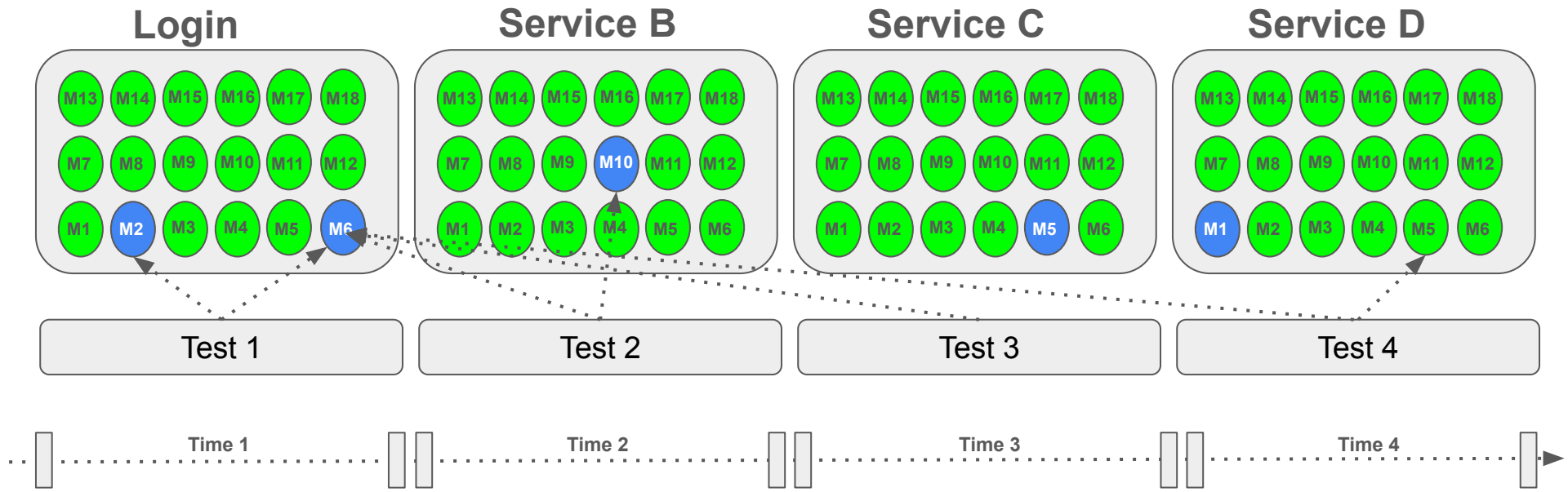
# Backup Slides

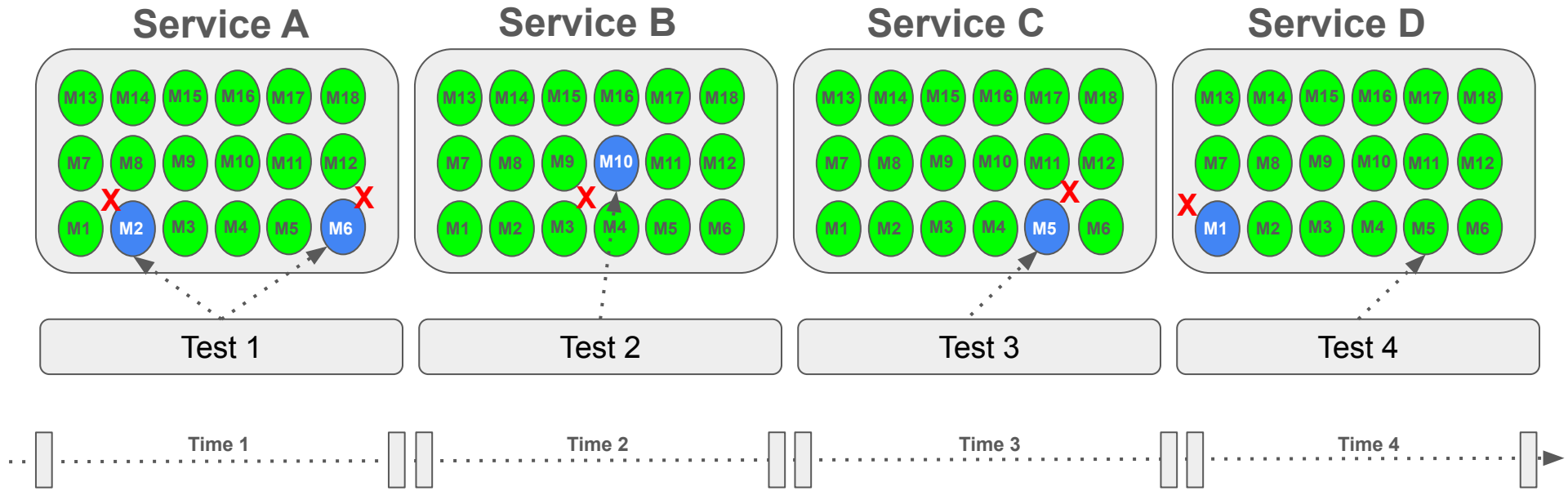
TIA Training

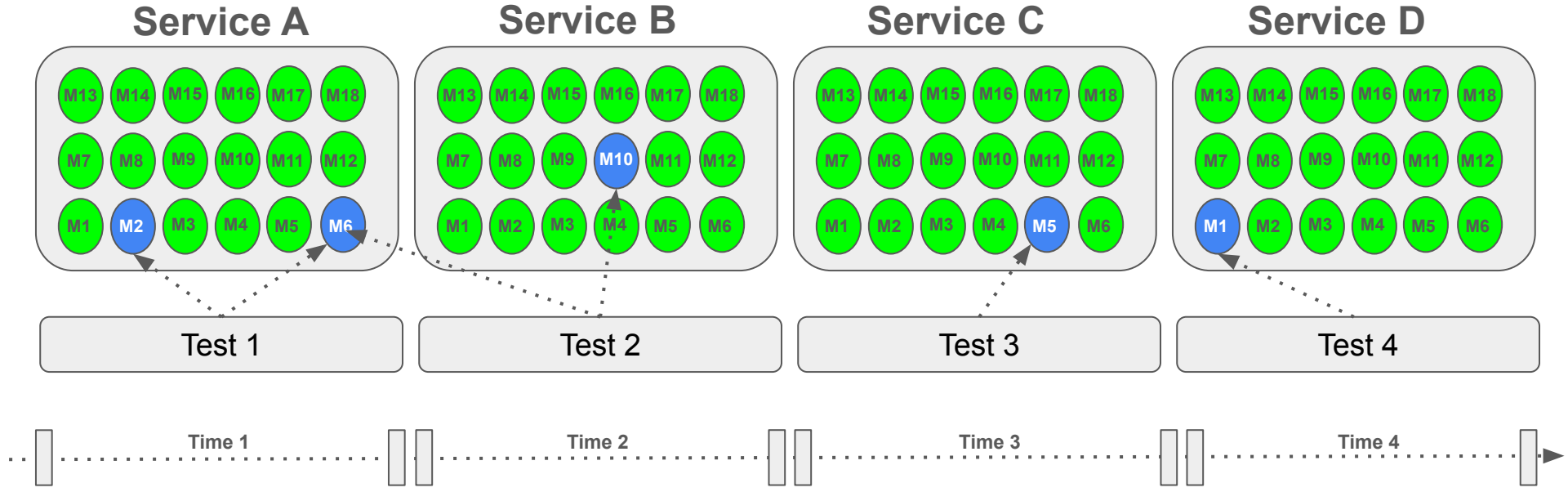


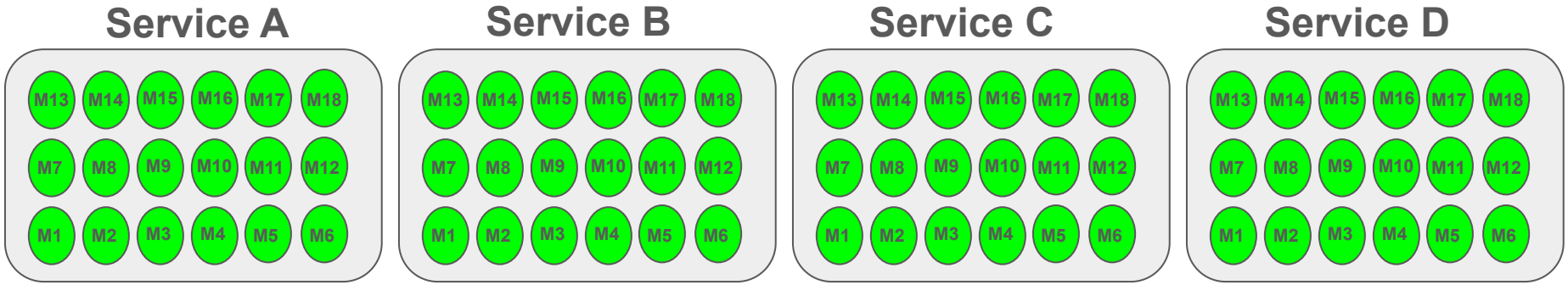
# Thank You

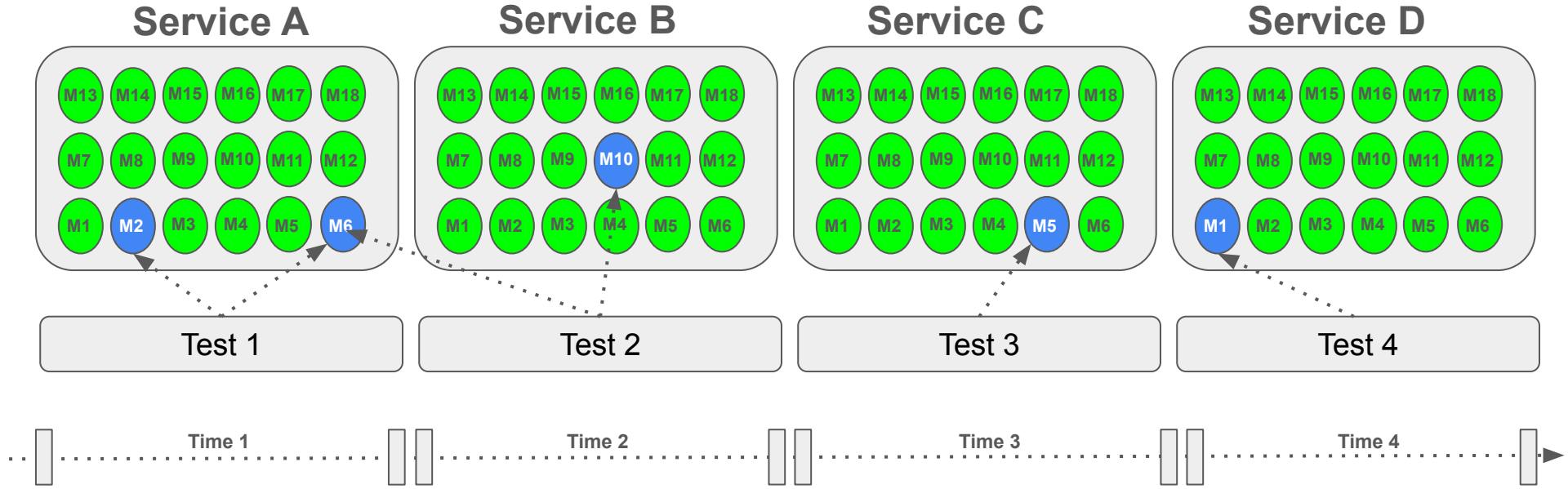
# Original Source


Login
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4
Time 1
Time 2
Time 3
Time 4

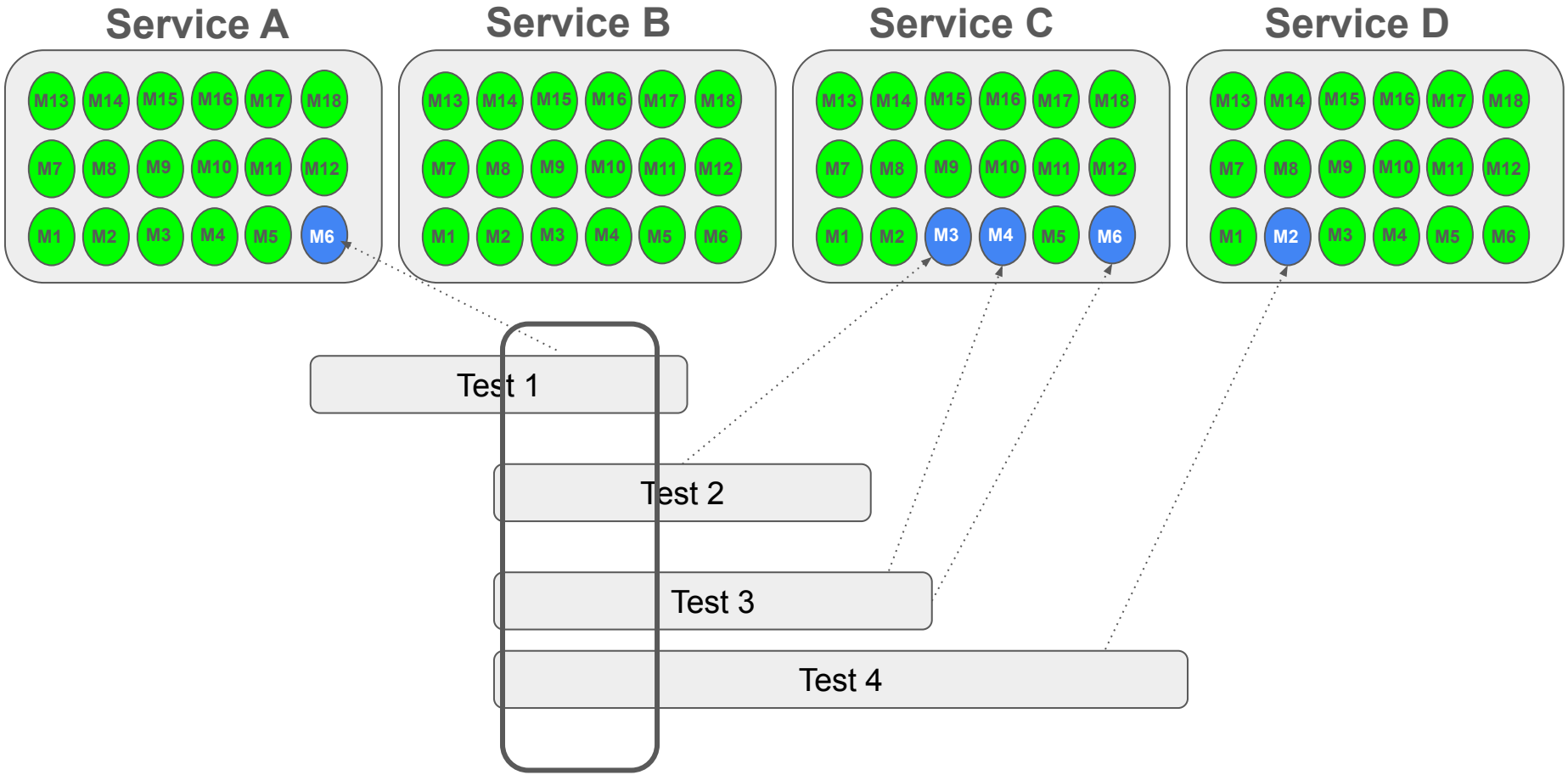

Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
X
Test 1
Test 2
Test 3
Test 4
Time 1
Time 2
Time 3
Time 4

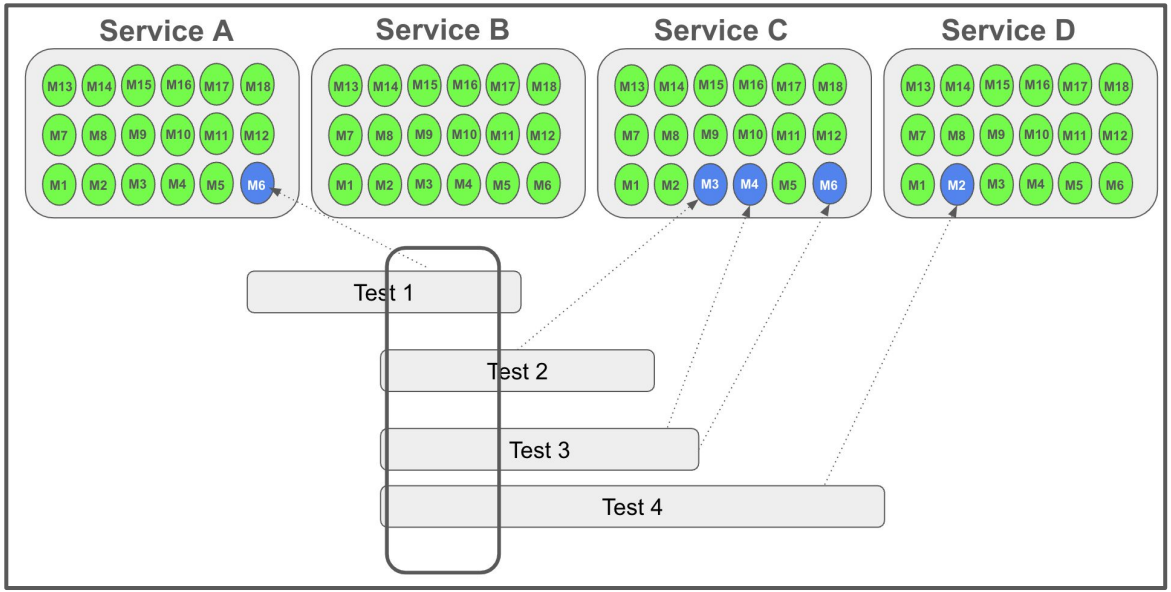

Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4
Time 1
Time 2
Time 3
Time 4


Service A
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Service B
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Service C
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6




| Service | Method | Tests |
|---|---|---|
| A | M2 | Test 1 |
| A | M6 | Test 1, Test 2 |
| B | M10 | Test 2 |
| C | M5 | Test 3 |
| D | M1 | Test 4 |


Service A
Service B
Service C
Service D
M13 M14 M15 M16 M17 M18
M7 M8 M9 M10 M11 M12
M1 M2 M3 M4 M5 M6
Test 1
Test 2
Test 3
Test 4




**Test to code mapping**

| Service | Method | Tests |
|---|---|---|
| A | M6 | Test 1, Test 2, Test 3, Test 4 |
| C | M3 | Test 2 |
| C | M4 | Test 3 |
| C | M6 | Test 3 |
| D | M2 | Test 4 |