AIGenOps

Integrating Generative AI into Platform Engineering for Regulated Software

imola informatica

Riccardo Soro

DevOps Engineer

rsoro@imolainformatica.it

riccardo-soro

Nicolas Fantoni

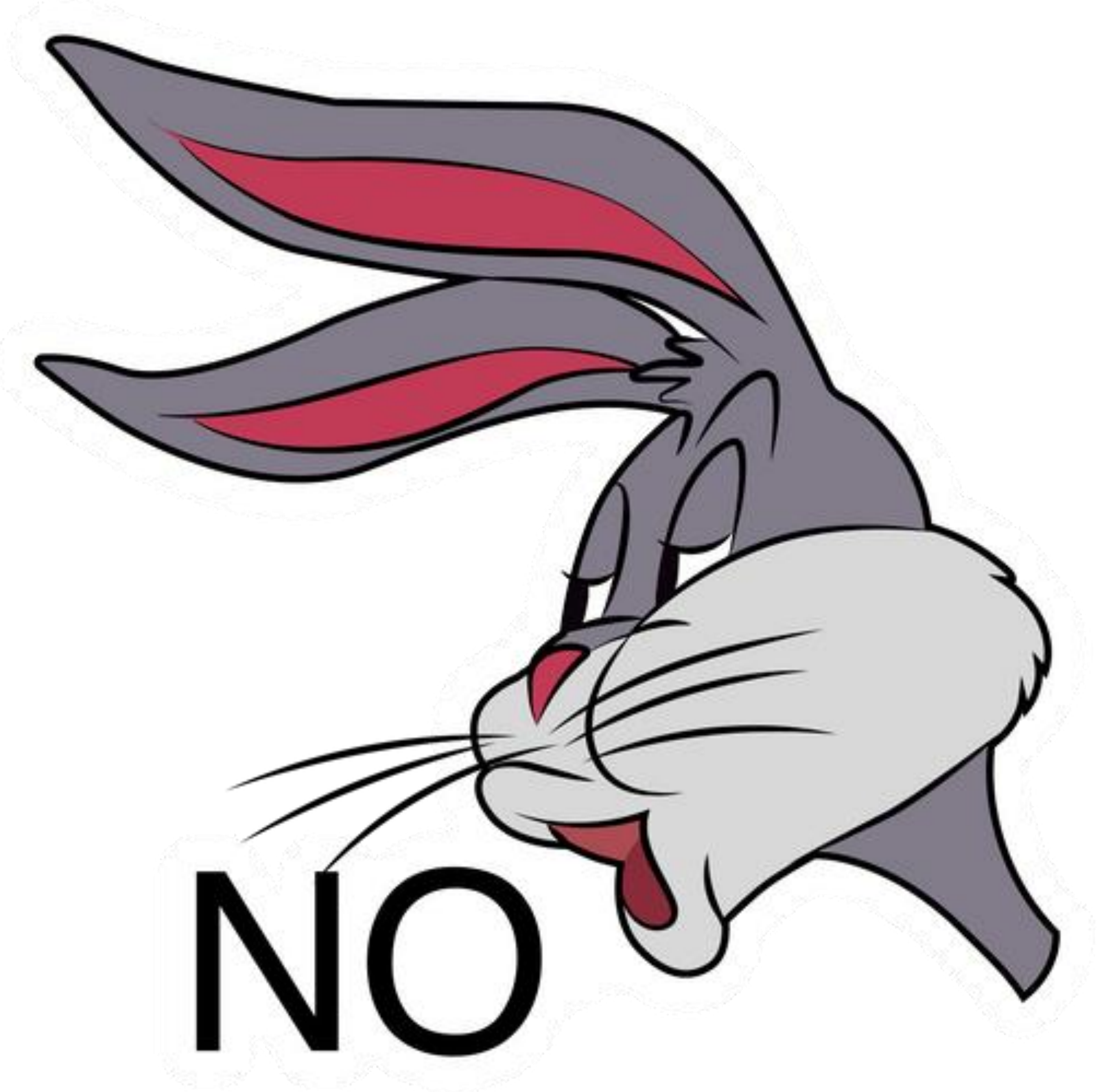
Platform Engineer

nfantoni@imolainformatica.it

nicolas_fantoni

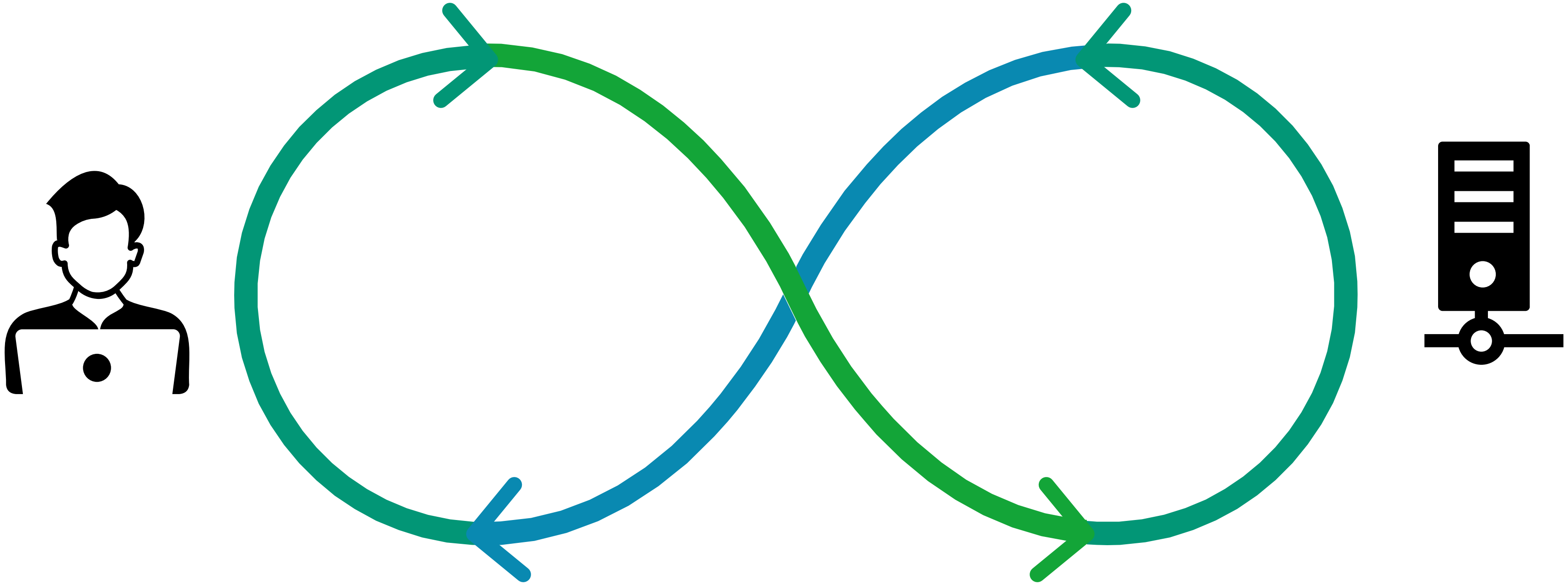
nicolas-fantoni

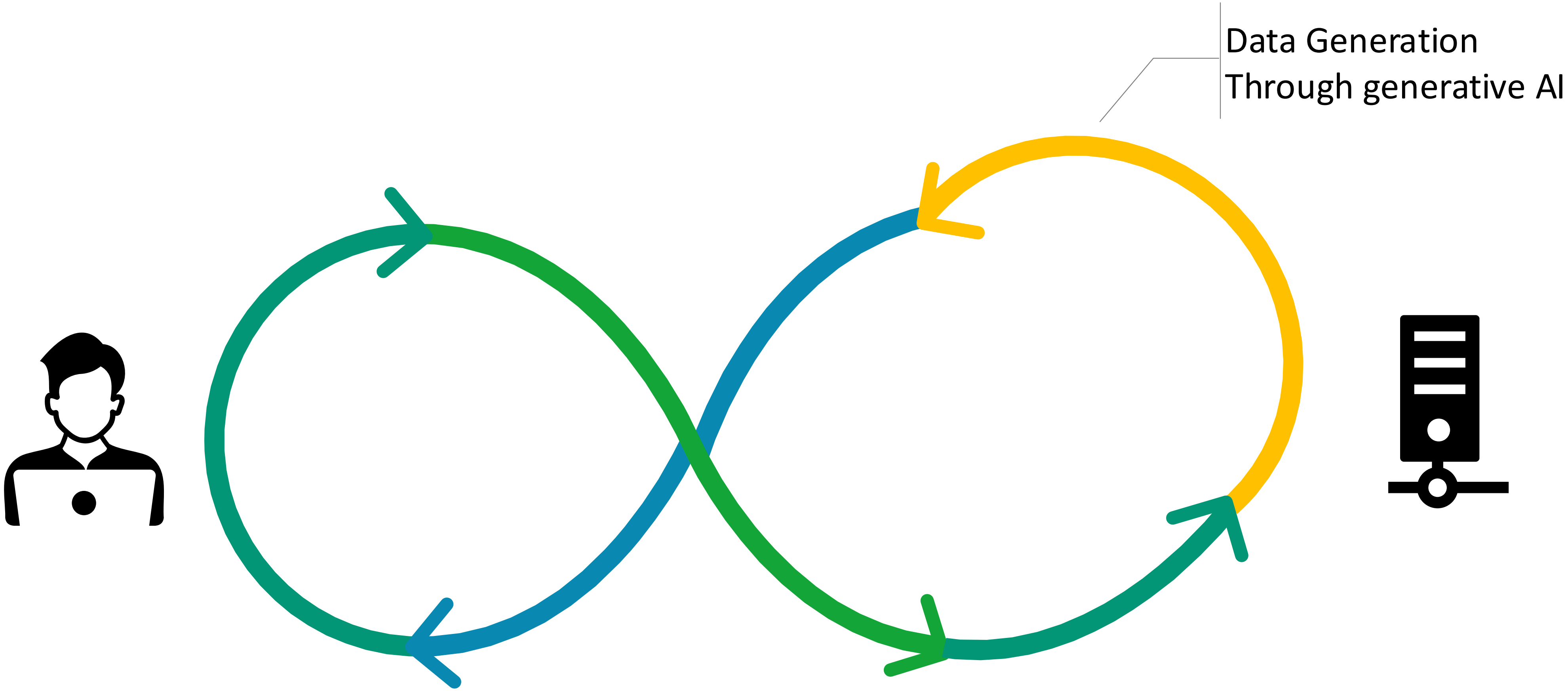
In the beginning...


NO

Requirements

- Performance
- Cost
- Security
- Quantity of generated data
- Zero Trust

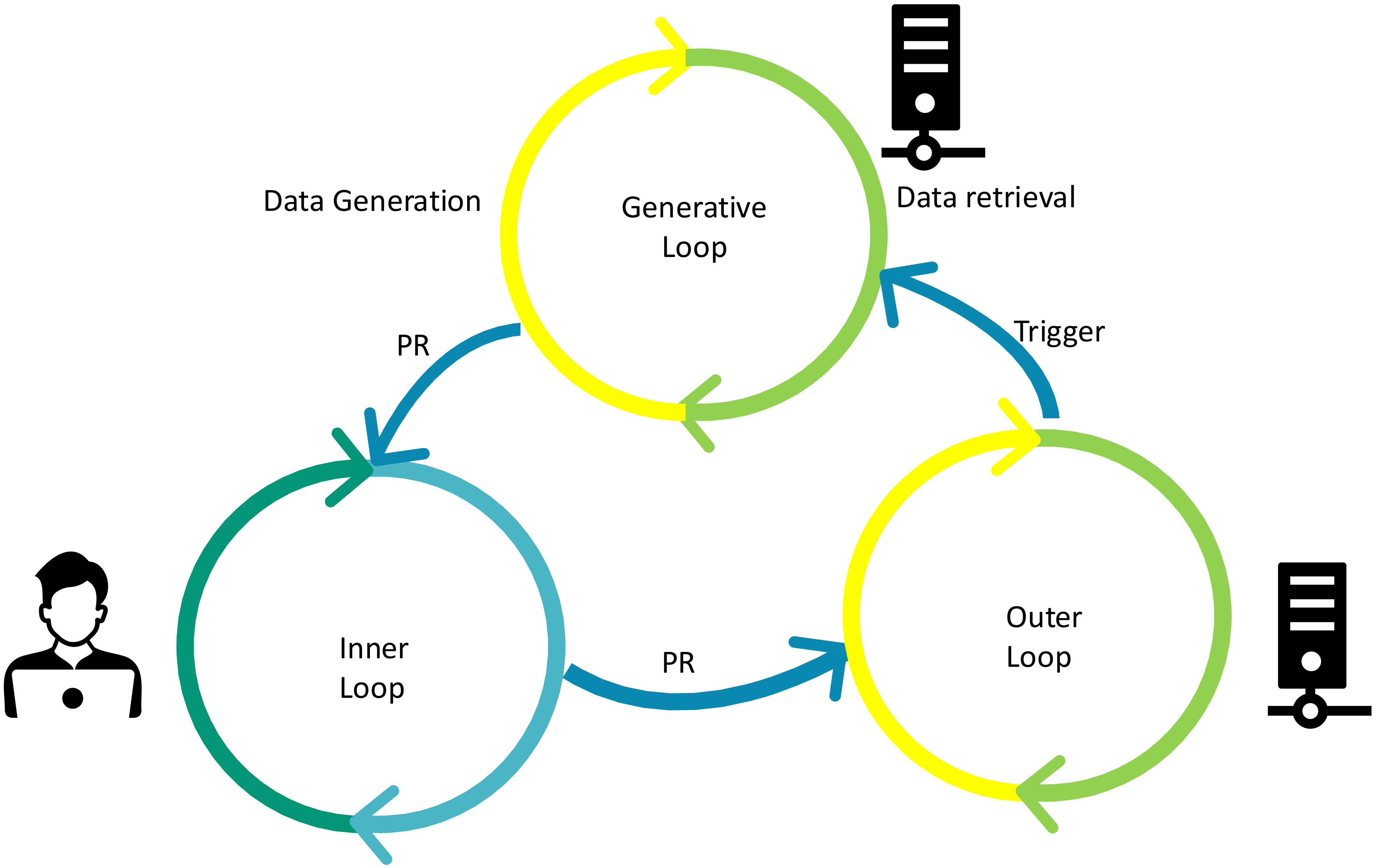




Data Generation
Through generative AI

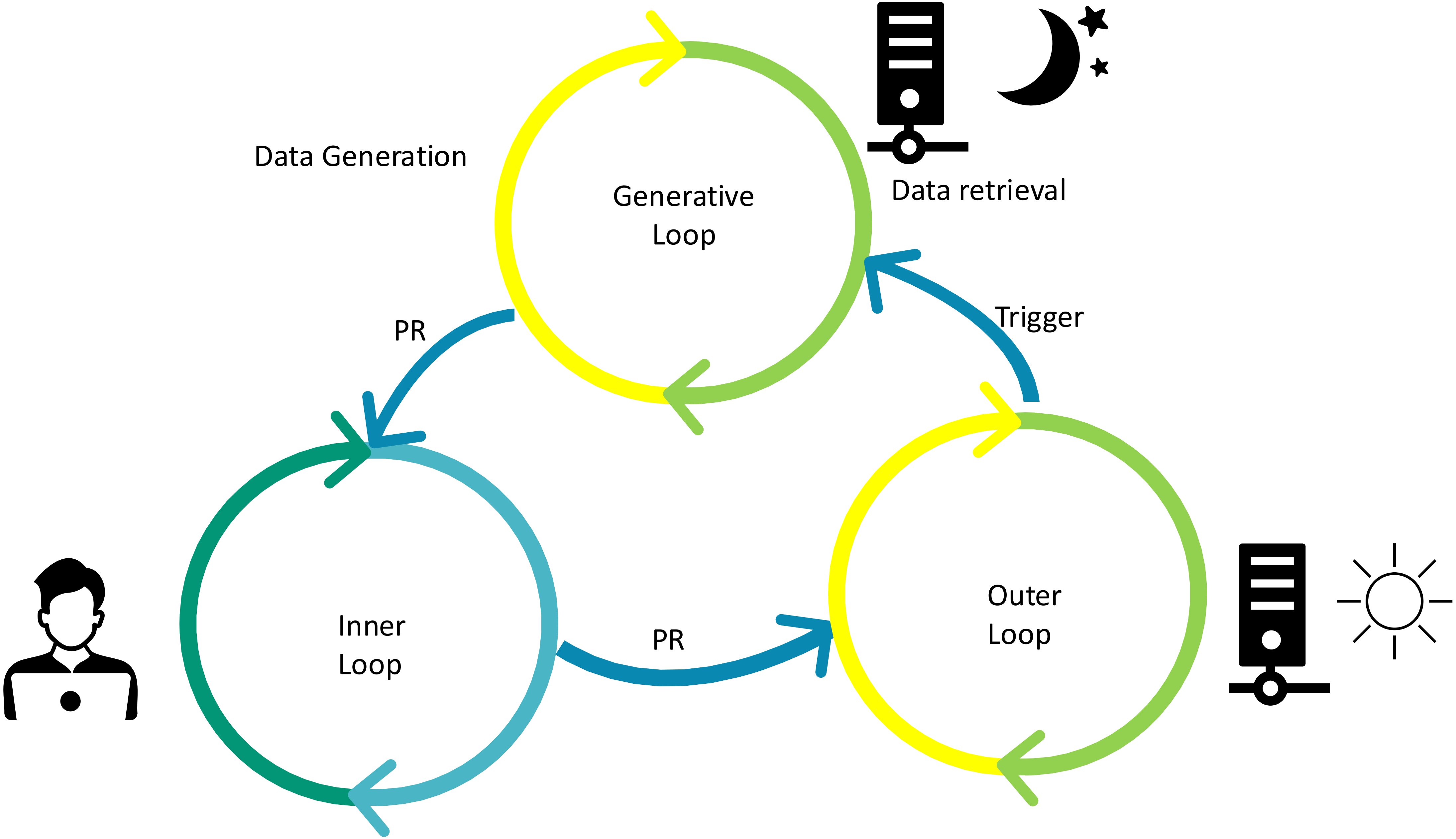
Requirements - Performance

Performance: avoiding degradation by introducing Gen AI

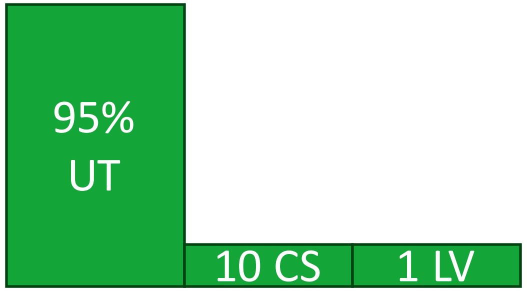

Data Generation
Generative Loop
Data retrieval
PR
Trigger
Inner Loop
PR
Outer Loop

Requirements - Cost

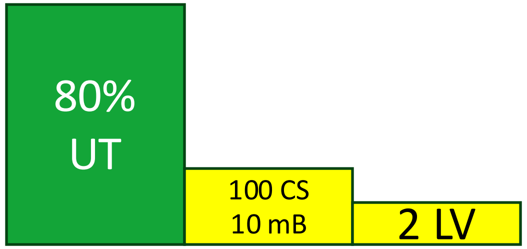
Cost: we do not live in a world of infinite resources


Data Generation
Generative
Loop
Data retrieval
PR
Trigger
Inner
Loop
PR
Outer
Loop

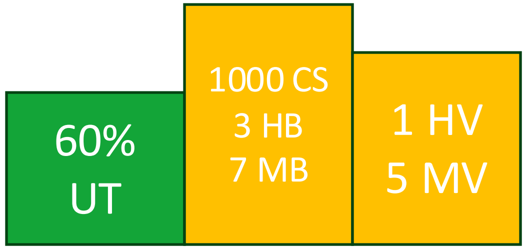
Application A



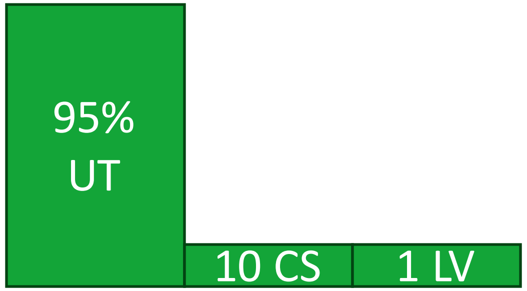
Application B



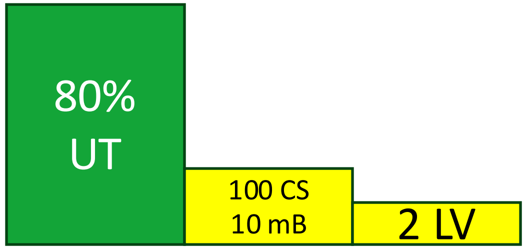
Application C



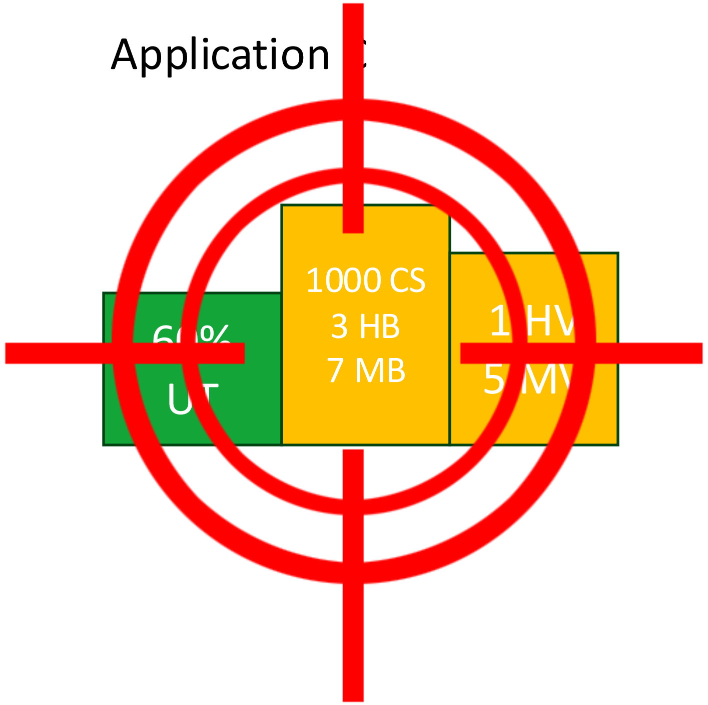
Application A

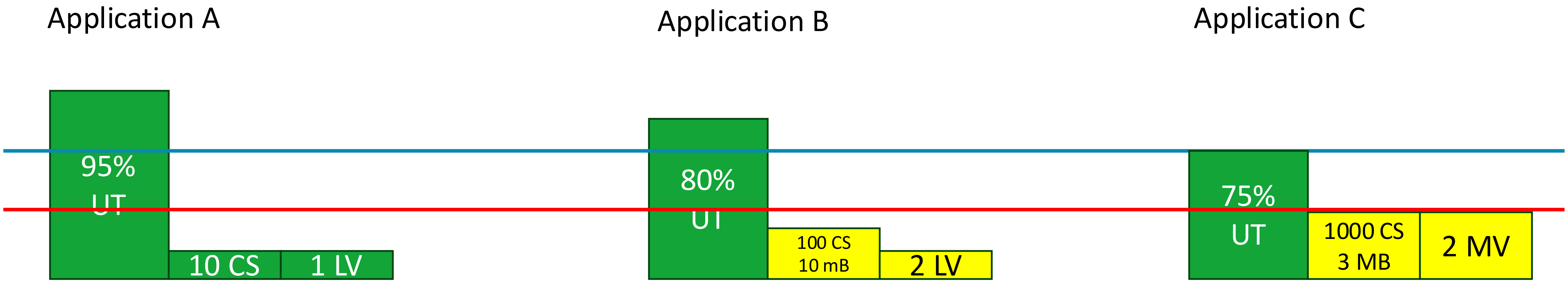


Application B



Application C





Application A

95%
UT

10 CS

1 LV

Application B

80%
UT

100 CS
10 mB

2 LV

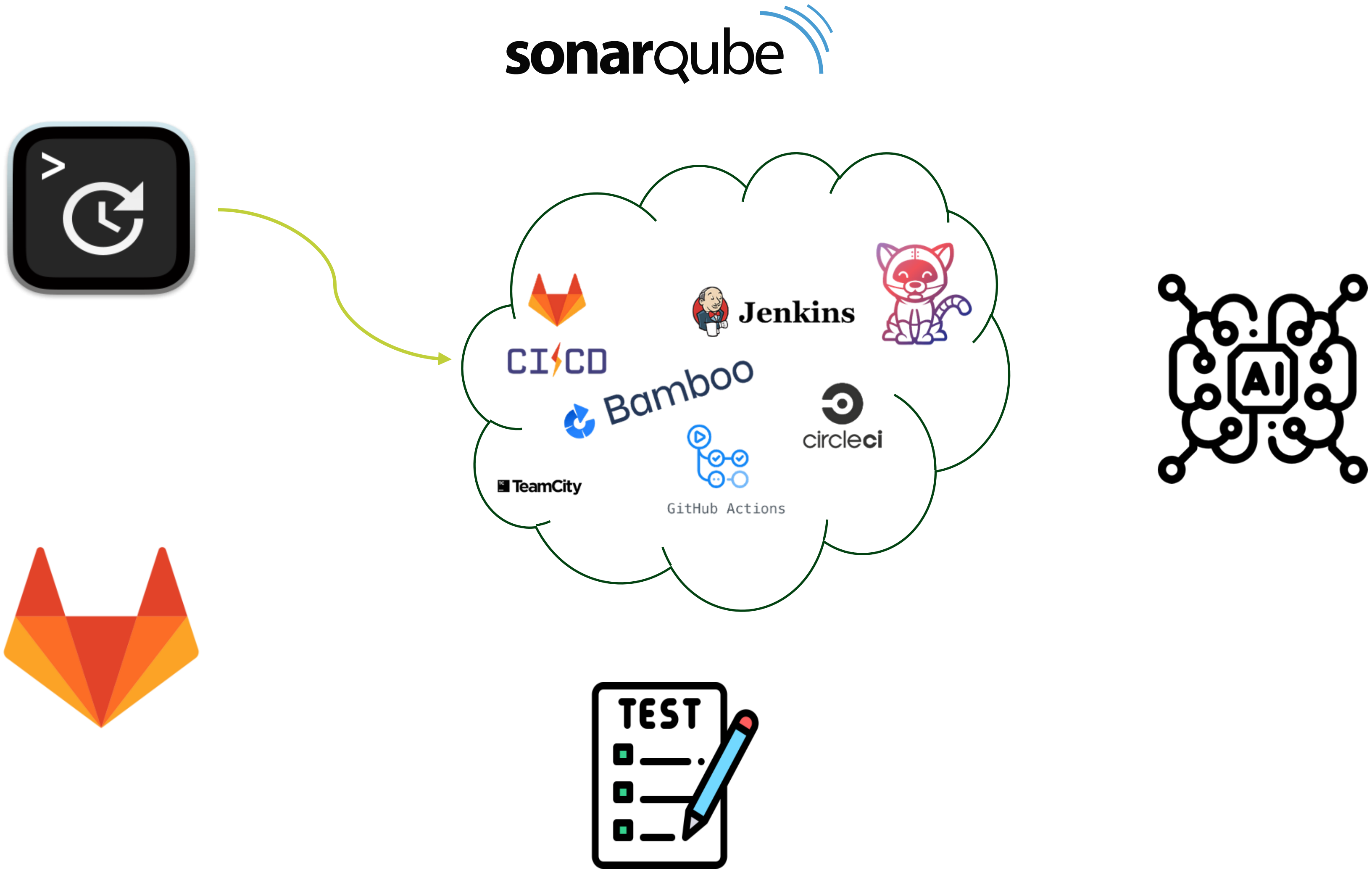
Application C

75%
UT

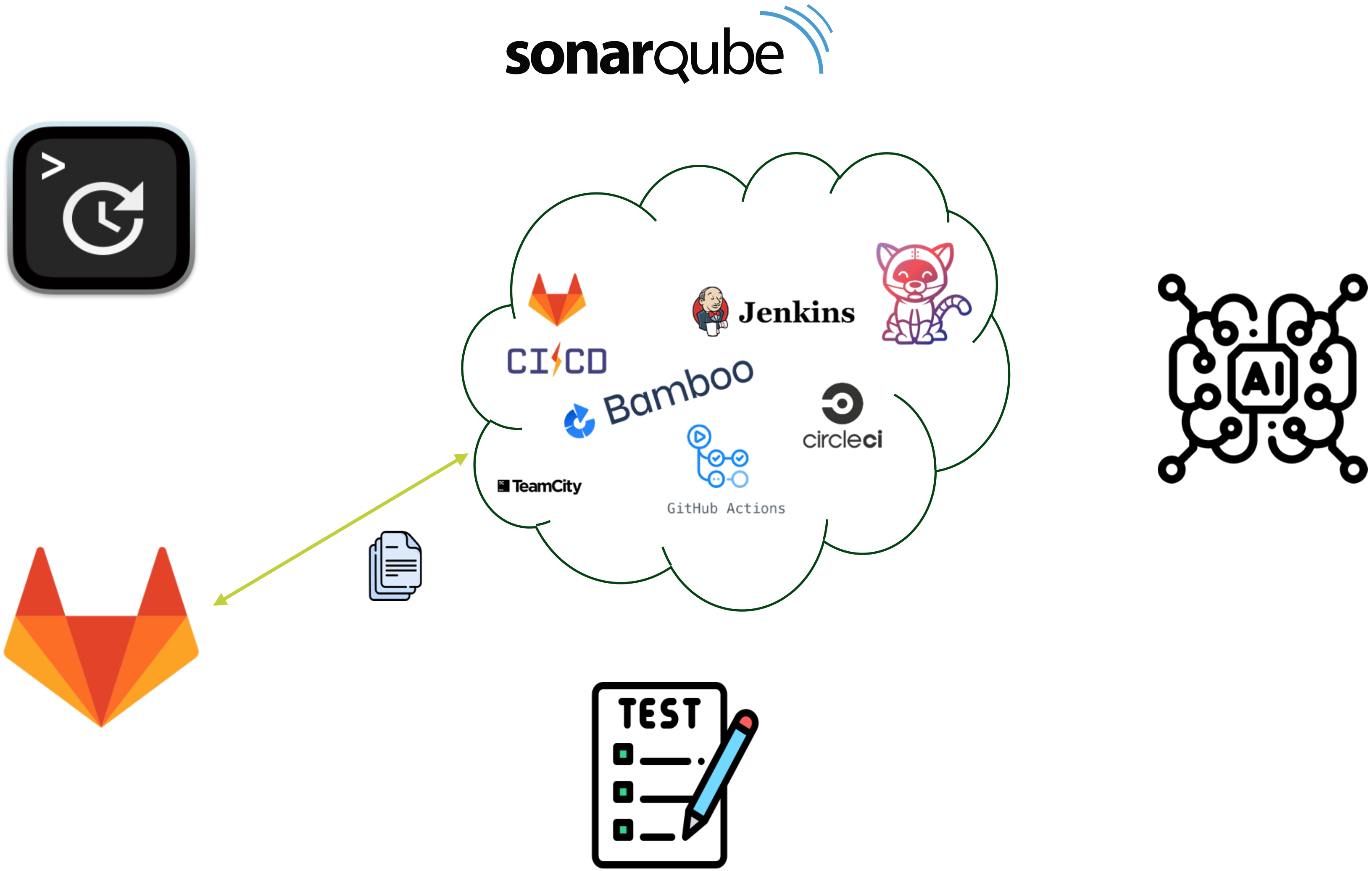
1000 CS
3 MB

2 MV

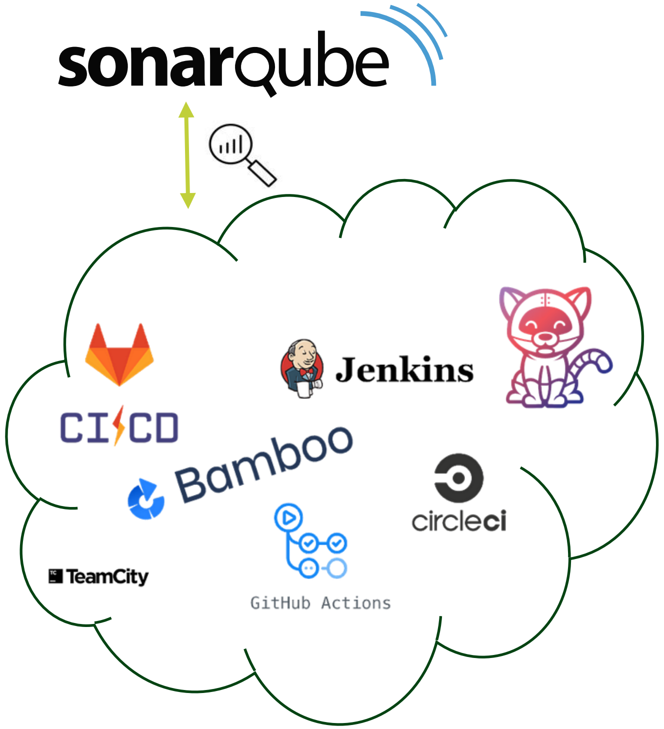
Integration with generative AI



Integration with generative AI

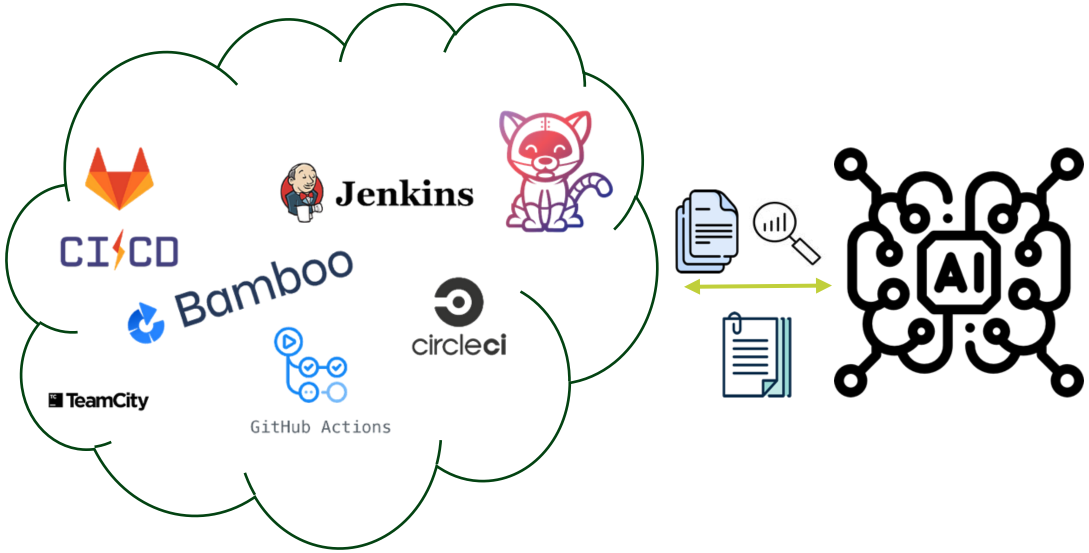


Integration with generative AI



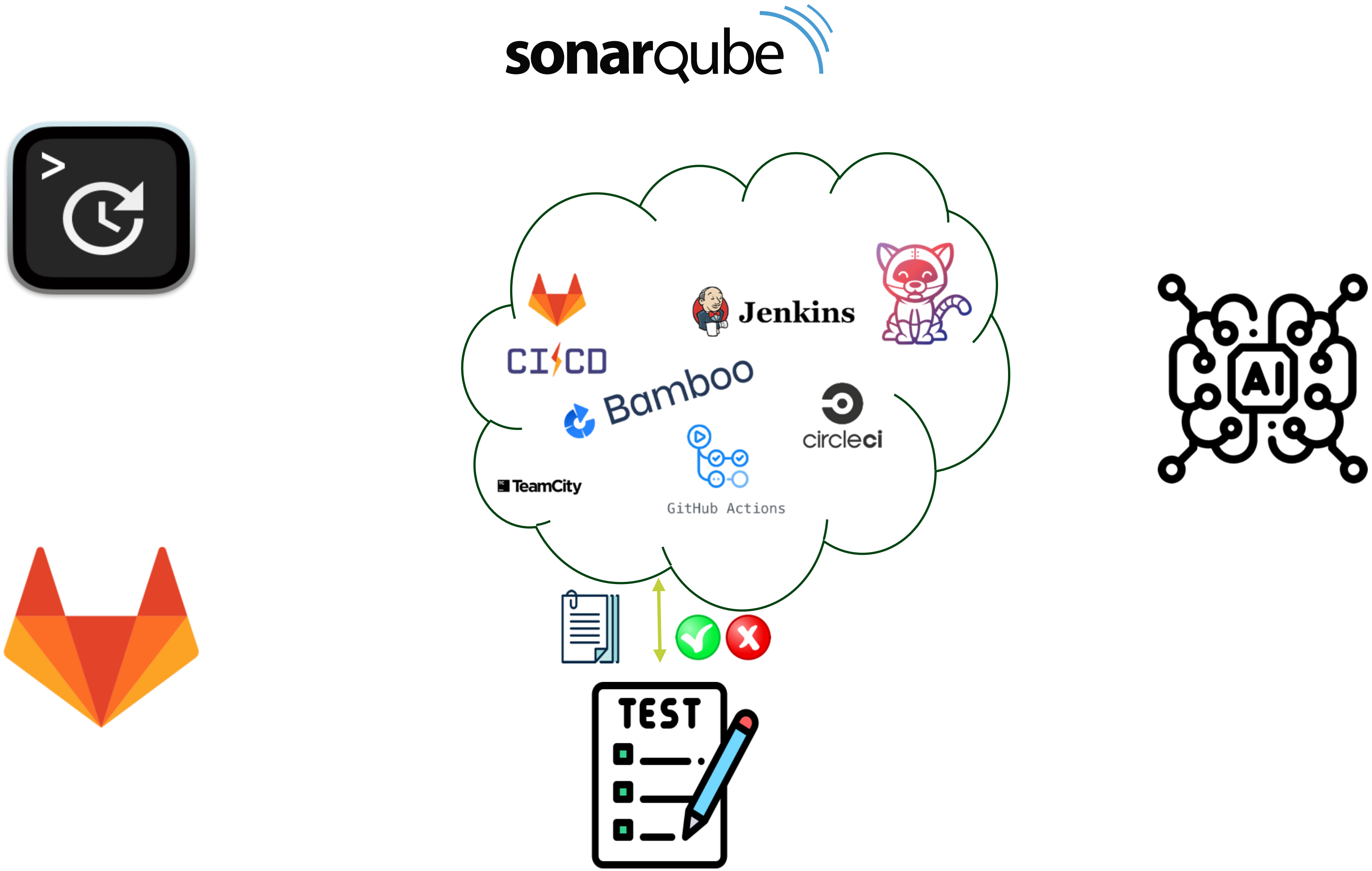
Integration with generative AI

sonarqube

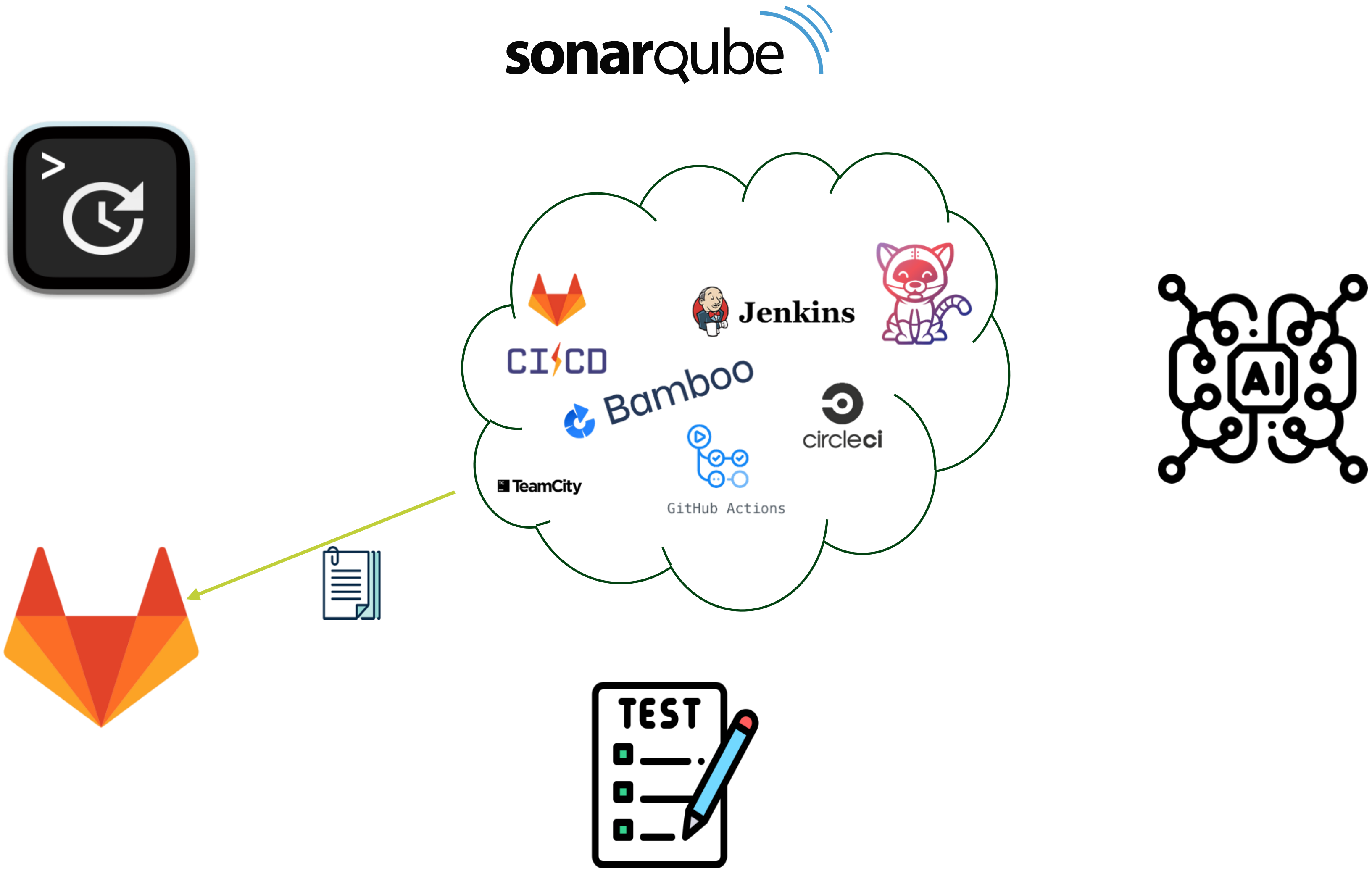


TEST

Integration with generative AI



Integration with generative AI



Which model to use?

ChatGPT

Gemini

Claude

Requirements - Sicurezza

Security: prefer self-hostable models

Which model to use?

~~ChatGPT~~

~~Gemini~~

~~Claude~~

MISTRAL AI_

code llama

databricks

Which model to use?

MISTRAL AI_

code llama

databricks

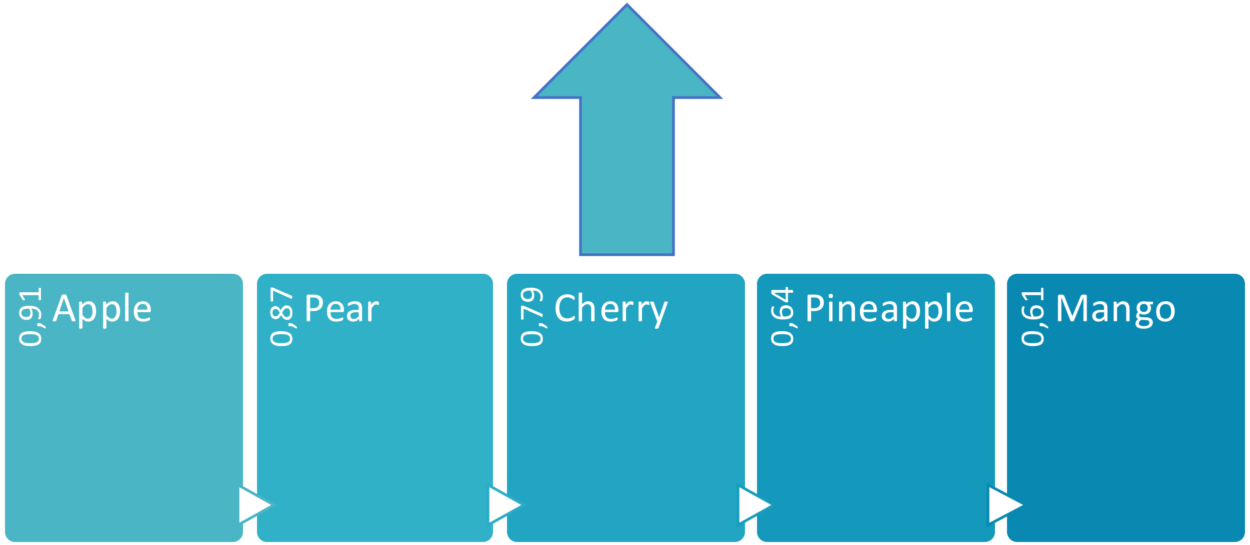
Hyperparameters

TOP-K
TOP-P
TEMPETATURE
PROMPT

Generative AI, introduction

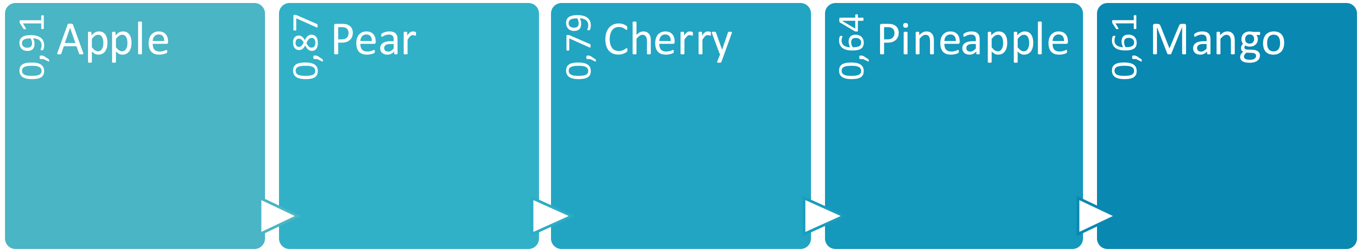
Question: What fruit can I put in the fruit salad?

Answer: in the fruit salad you can put...



Hyperparameters: Top-K

- K = 5



0,91 Apple

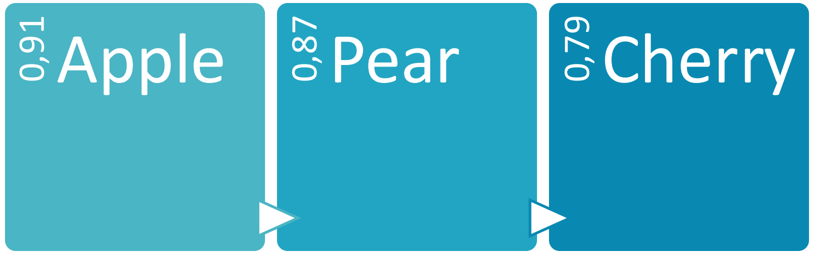
0,87 Pear

0,79 Cherry

0,64 Pineapple

0,61 Mango

- K = 3



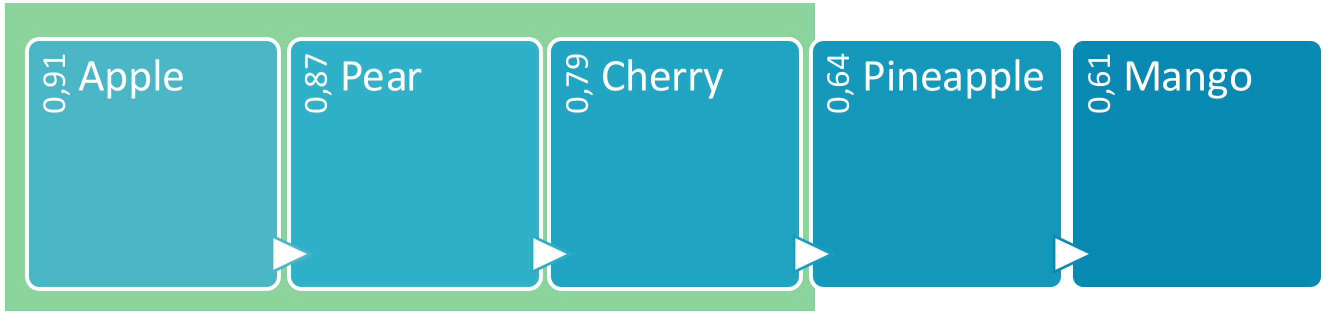
0,91 Apple

0,87 Pear

0,79 Cherry

Hyperparameters: Top-P

- P = 2



0,91 Apple

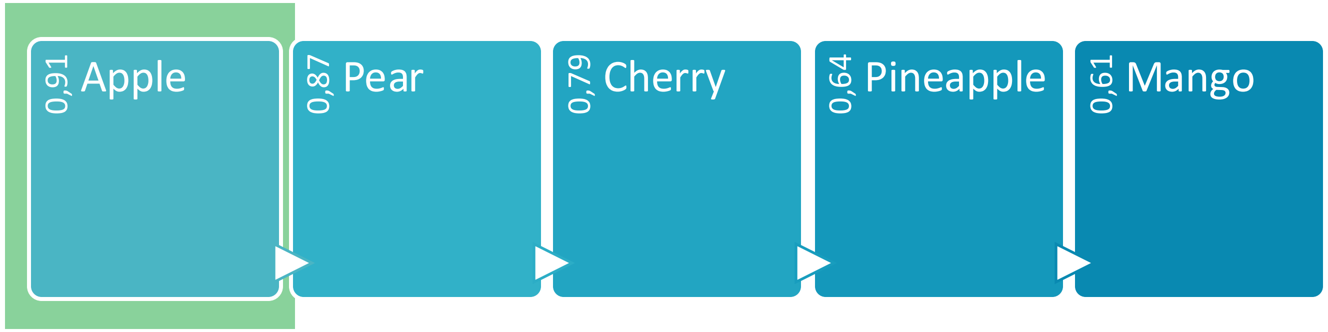
0,87 Pear

0,79 Cherry

0,64 Pineapple

0,61 Mango

- P = 0.8



0,91 Apple

0,87 Pear

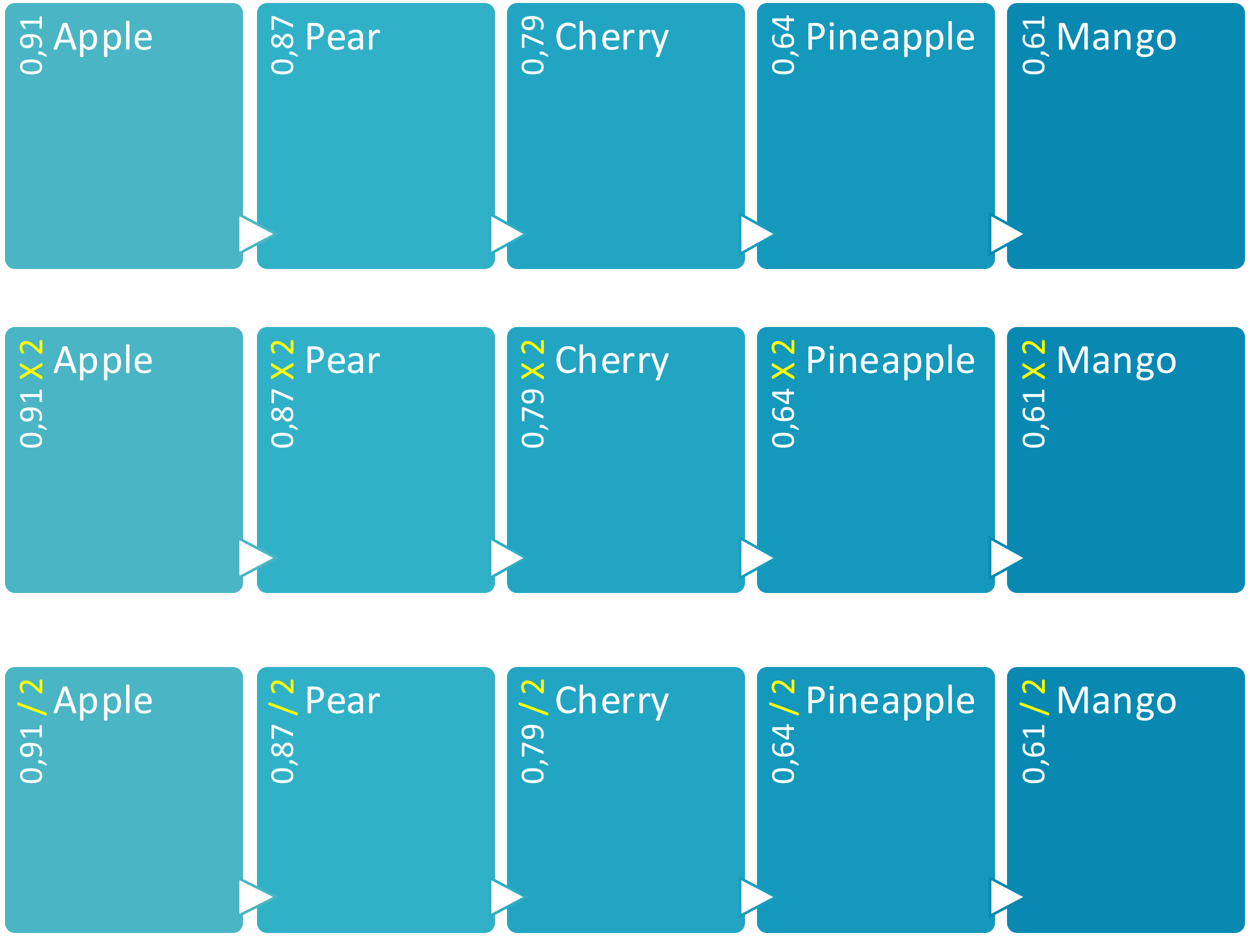
0,79 Cherry

0,64 Pineapple

0,61 Mango

Hyperparameters: Temperature

- T = 1
- T = 0.5
- T = 2



0,91 Apple	0,87 Pear	0,79 Cherry	0,64 Pineapple	0,61 Mango
0,91 X 2 Apple	0,87 X 2 Pear	0,79 X 2 Cherry	0,64 X 2 Pineapple	0,61 X 2 Mango
0,91 / 2 Apple	0,87 / 2 Pear	0,79 / 2 Cherry	0,64 / 2 Pineapple	0,61 / 2 Mango

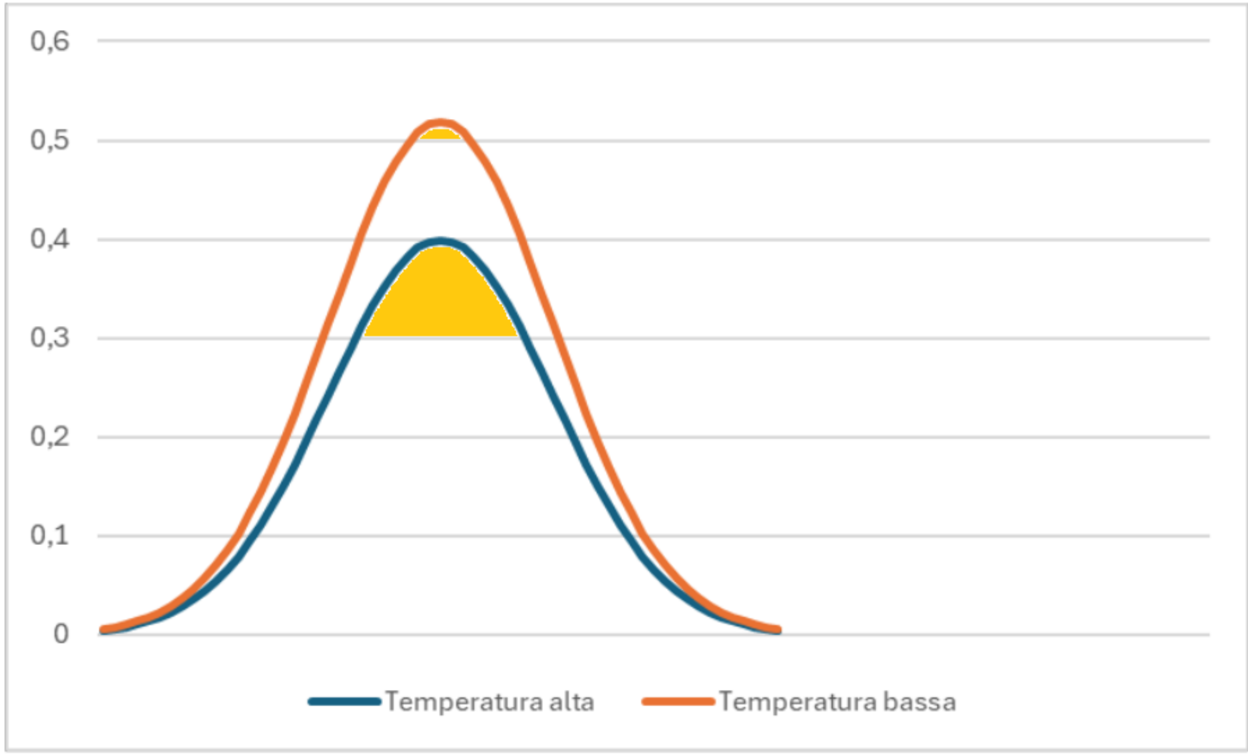
Hyperparameters: Temperature

Considerations

High temperature = greater randomness in response

Temperature at 0 = deterministic model (approximately), local optimum is sought

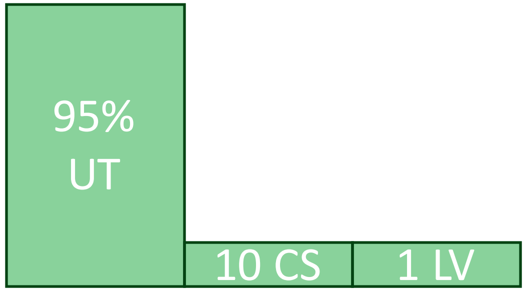
Reiterate on high temperature finds global optimum sooner or later



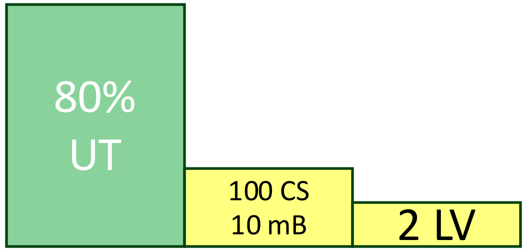
Requirements - Quantity of generated data

Quantity of generated data: we have to limit the amount of data generated

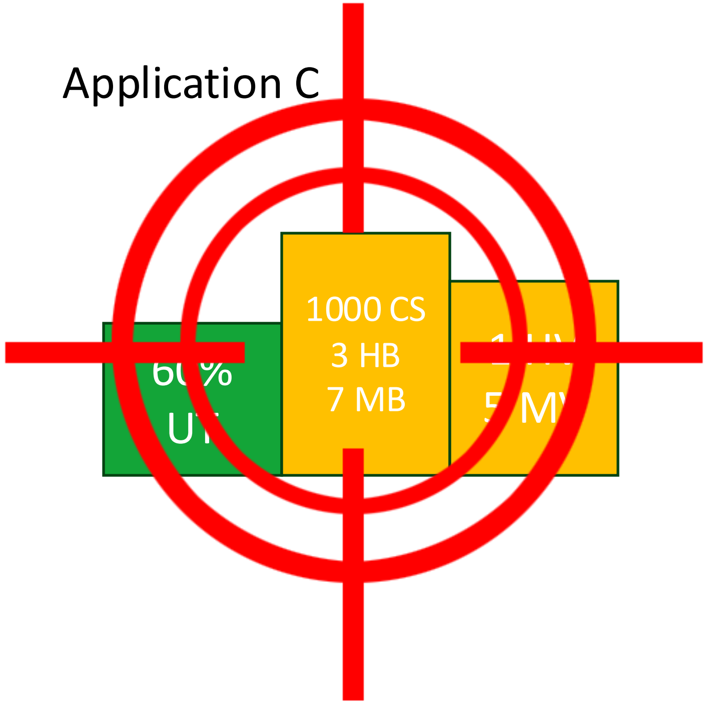
Application A



Application B



Application C



Prompt Template

Focus on context

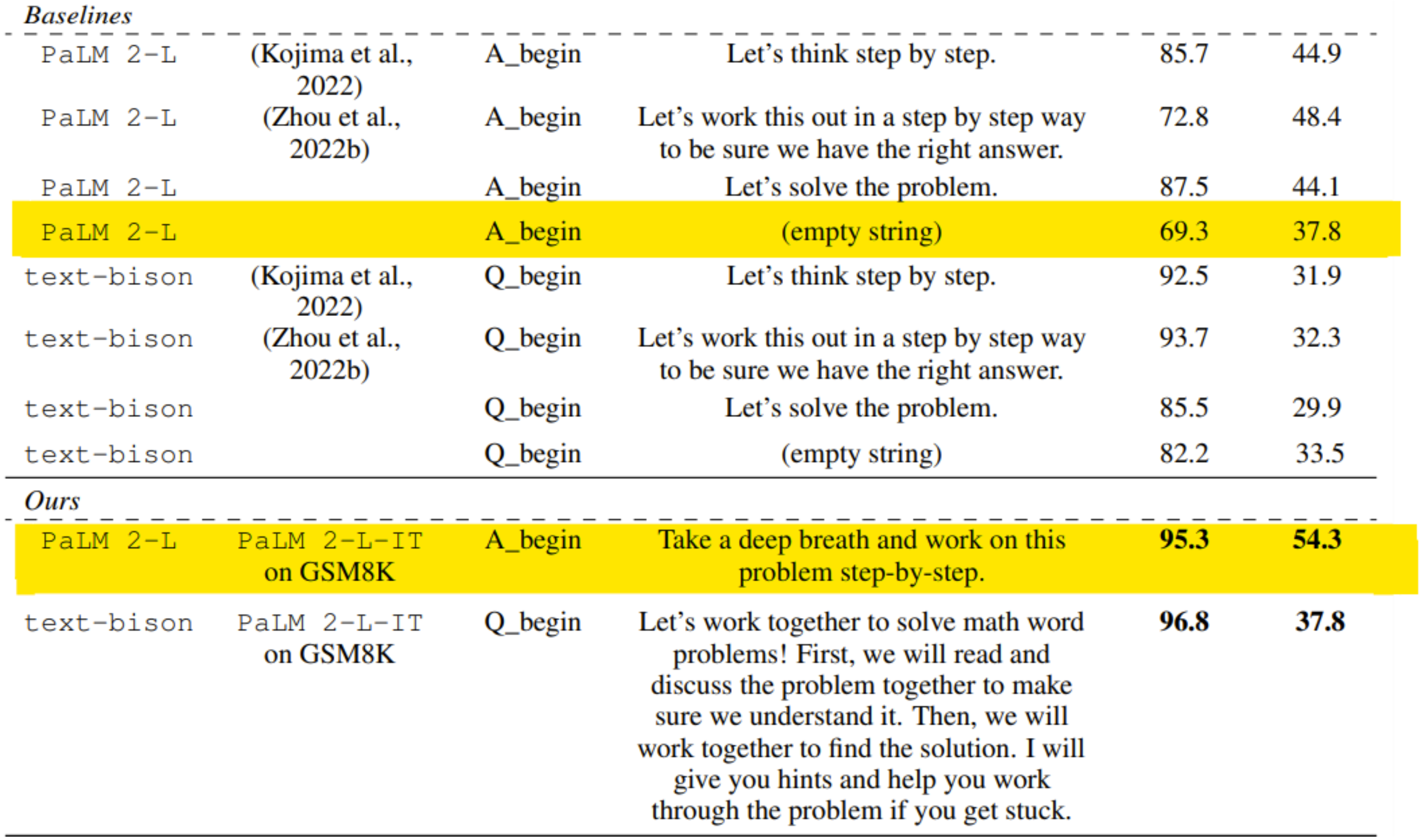
### Instructions:

Forget everything you know and by analysing the code I will pass on to you solve this issue: {issue}

{code}

### Response:

reduction of hallucinations



Baselines					
PaLM 2-L	(Kojima et al., 2022)	A_begin	Let's think step by step.	85.7	44.9
PaLM 2-L	(Zhou et al., 2022b)	A_begin	Let's work this out in a step by step way to be sure we have the right answer.	72.8	48.4
PaLM 2-L		A_begin	Let's solve the problem.	87.5	44.1
PaLM 2-L		A_begin	(empty string)	69.3	37.8
text-bison	(Kojima et al., 2022)	Q_begin	Let's think step by step.	92.5	31.9
text-bison	(Zhou et al., 2022b)	Q_begin	Let's work this out in a step by step way to be sure we have the right answer.	93.7	32.3
text-bison		Q_begin	Let's solve the problem.	85.5	29.9
text-bison		Q_begin	(empty string)	82.2	33.5
Ours					
PaLM 2-L	PaLM 2-L-IT on GSM8K	A_begin	Take a deep breath and work on this problem step-by-step.	**95.3**	**54.3**
text-bison	PaLM 2-L-IT on GSM8K	Q_begin	Let's work together to solve math word problems! First, we will read and discuss the problem together to make sure we understand it. Then, we will work together to find the solution. I will give you hints and help you work through the problem if you get stuck.	**96.8**	**37.8**

Requirements - Zero Trust

Zero Trust: the genereted code must be checked

How do you test the generated Unit Tests?

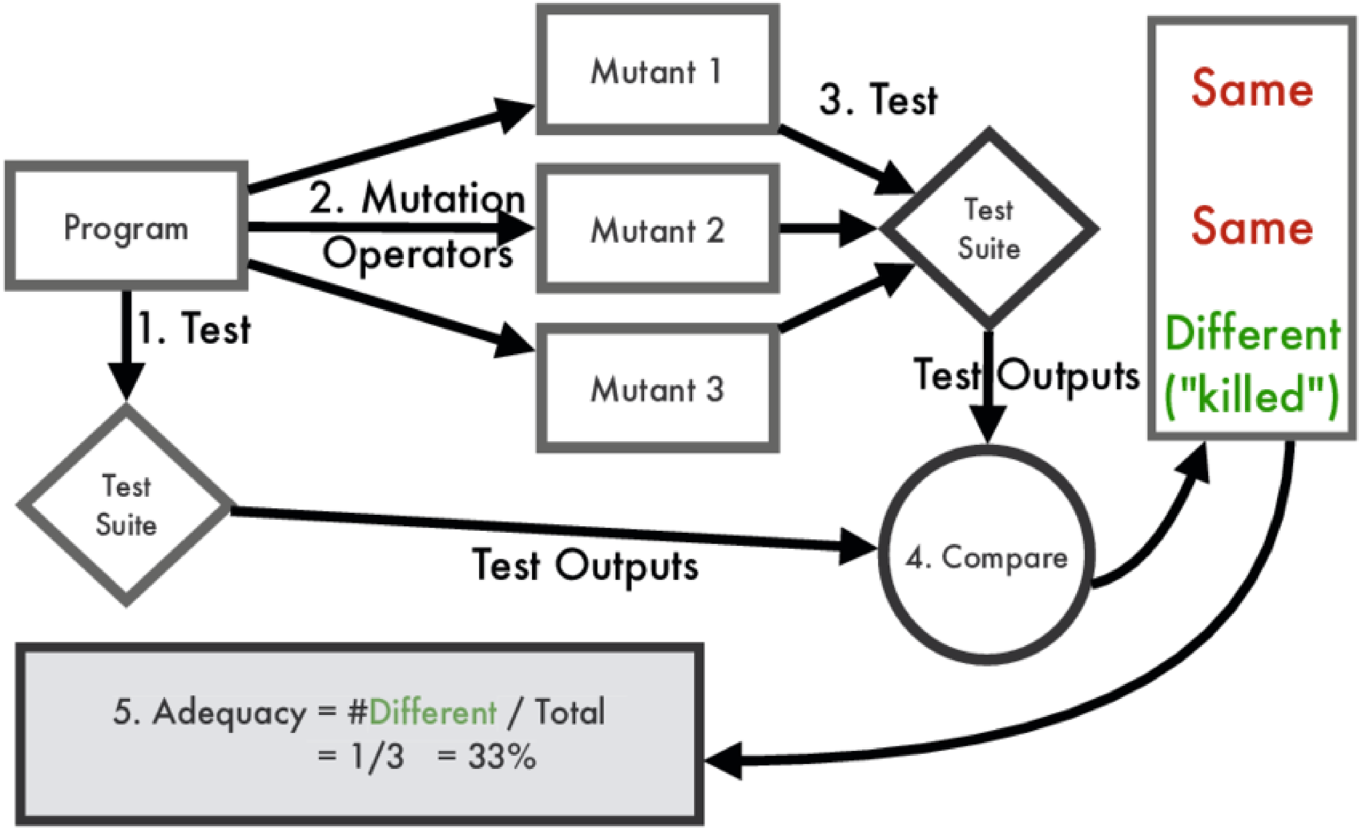
Validation requirement of the generated Unit Test.

- The project must compile
- Tests must pass
- Coverage must increase
- Unit Test must be of quality

However, how do you define a quality Unit Test?

Use of mutation tests

What are mutation tests?



Idea:

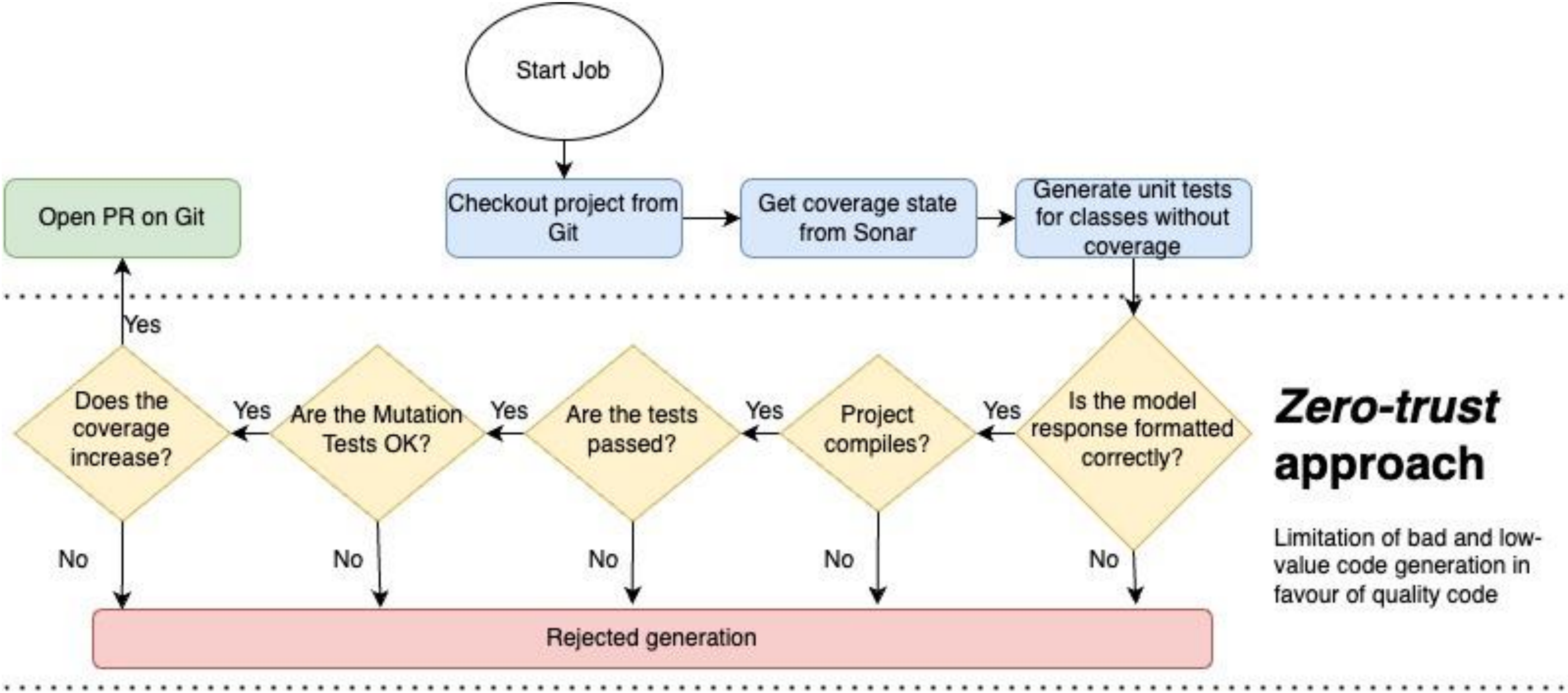
A quality Unit Test must fail if the logic in the code changes (e.g. the conditions of an if are changed)

How do you test fixes to QA Issues?

Validation requirement of the fix generated:

- The project must compile
- Tests must pass
  - Coverage must be high
  - Tests must be of quality
- The generated code must solve the issue and must not introduce others

Flow chart: unit test generation

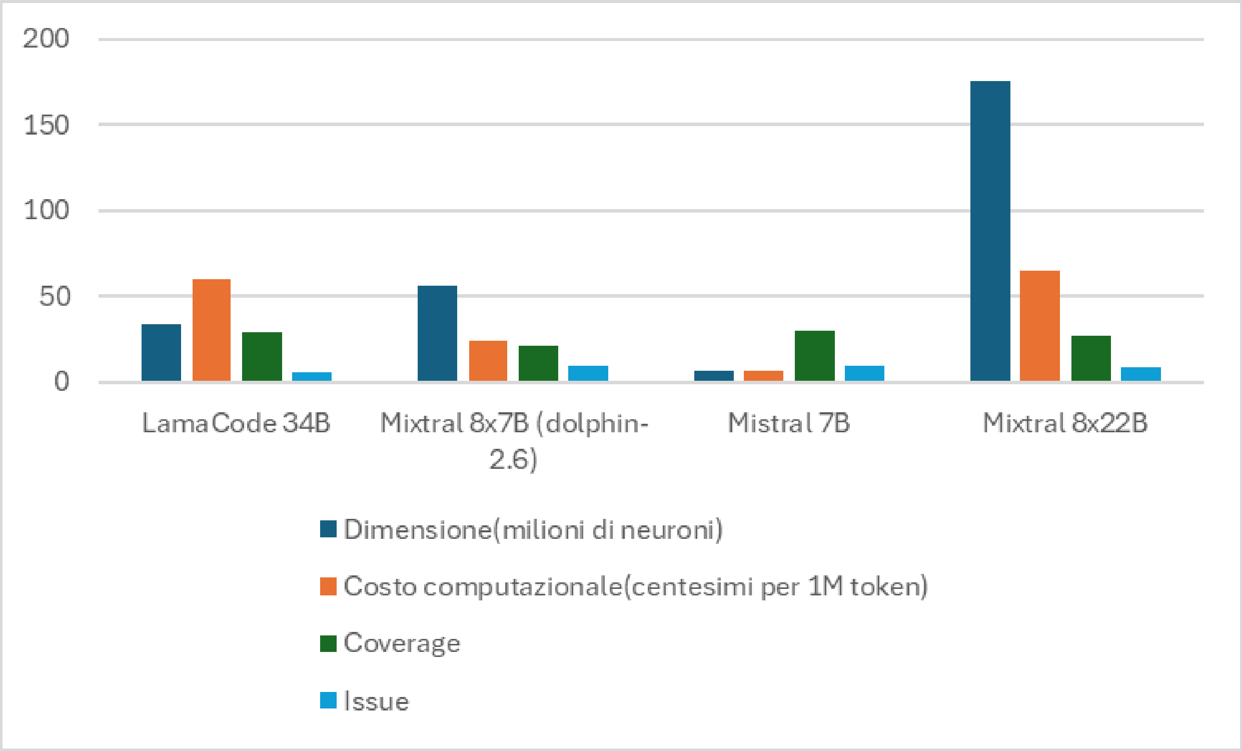


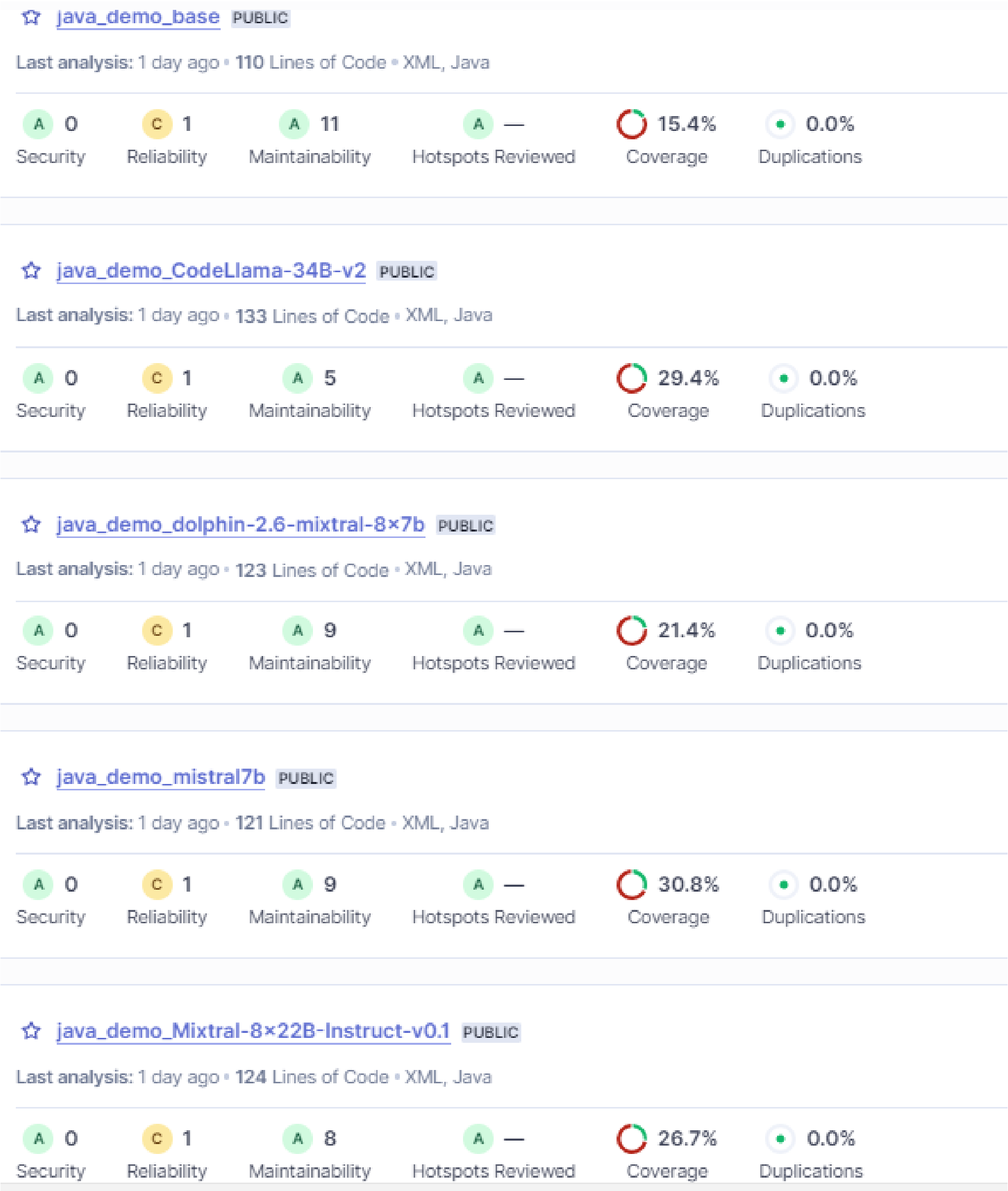
Start Job
Checkout project from Git
Get coverage state from Sonar
Generate unit tests for classes without coverage
Open PR on Git
Yes
Is the model response formatted correctly?
Yes
Project compiles?
Yes
Are the tests passed?
Yes
Are the Mutation Tests OK?
Yes
Does the coverage increase?
No
No
No
No
No
Rejected generation

Zero-trust approach

Limitation of bad and low-value code generation in favour of quality code

Results





java_demo_base PUBLIC

Last analysis: 1 day ago • 110 Lines of Code • XML, Java

A 0 Security · C 1 Reliability · A 11 Maintainability · A — Hotspots Reviewed · 15.4% Coverage · 0.0% Duplications

java_demo_CodeLlama-34B-v2 PUBLIC

Last analysis: 1 day ago • 133 Lines of Code • XML, Java

A 0 Security · C 1 Reliability · A 5 Maintainability · A — Hotspots Reviewed · 29.4% Coverage · 0.0% Duplications

java_demo_dolphin-2.6-mixtral-8×7b PUBLIC

Last analysis: 1 day ago • 123 Lines of Code • XML, Java

A 0 Security · C 1 Reliability · A 9 Maintainability · A — Hotspots Reviewed · 21.4% Coverage · 0.0% Duplications

java_demo_mistral7b PUBLIC

Last analysis: 1 day ago • 121 Lines of Code • XML, Java

A 0 Security · C 1 Reliability · A 9 Maintainability · A — Hotspots Reviewed · 30.8% Coverage · 0.0% Duplications

java_demo_Mixtral-8×22B-Instruct-v0.1 PUBLIC

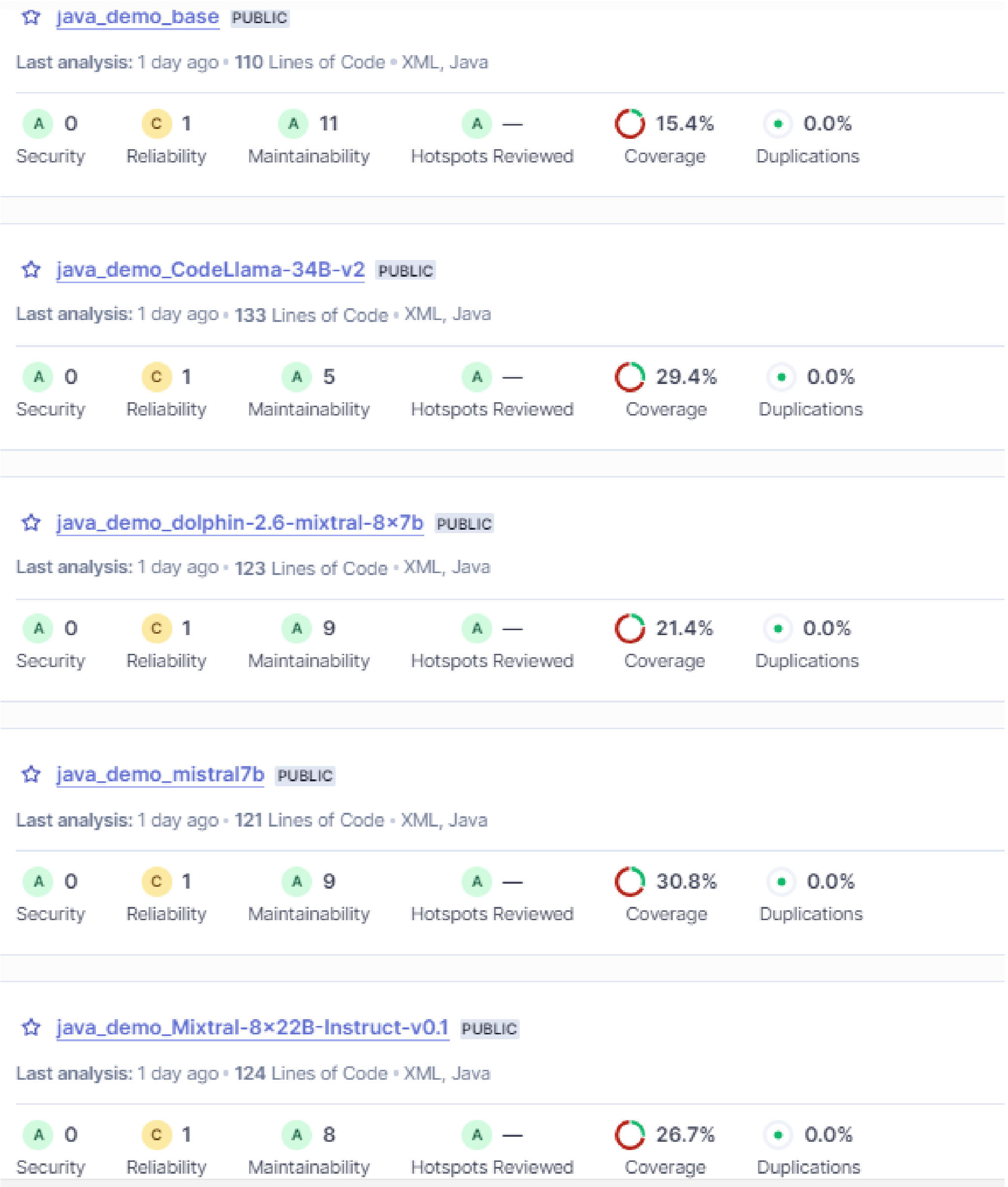
Last analysis: 1 day ago • 124 Lines of Code • XML, Java

A 0 Security · C 1 Reliability · A 8 Maintainability · A — Hotspots Reviewed · 26.7% Coverage · 0.0% Duplications

Considerations

- Generating fixes is more complex than generating test classes
- Specialization better than Dimension
- Some models do not require a GPU (mistral7B)

Model	Dimension	Resolved Issue	Coverage generated
LamaCode 34B	34	6	14%
Dolphin-2.6-mixtral 8x7B	56	2	6%
Mistral 7B	7	2	15.4%
Mixtral 8X22B	176	3	9,3%



☆ java_demo_base PUBLIC

Last analysis: 1 day ago • 110 Lines of Code • XML, Java

A 0	C 1	A 11	A —	15.4%	0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

☆ java_demo_CodeLlama-34B-v2 PUBLIC

Last analysis: 1 day ago • 133 Lines of Code • XML, Java

A 0	C 1	A 5	A —	29.4%	0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

☆ java_demo_dolphin-2.6-mixtral-8×7b PUBLIC

Last analysis: 1 day ago • 123 Lines of Code • XML, Java

A 0	C 1	A 9	A —	21.4%	0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

☆ java_demo_mistral7b PUBLIC

Last analysis: 1 day ago • 121 Lines of Code • XML, Java

A 0	C 1	A 9	A —	30.8%	0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

☆ java_demo_Mixtral-8×22B-Instruct-v0.1 PUBLIC

Last analysis: 1 day ago • 124 Lines of Code • XML, Java

A 0	C 1	A 8	A —	26.7%	0.0%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

WHAT'S
NEXT?

- Collaboration with opensource projects
- Define the application prioritization and selection algorithms
- Define the selection and prioritization algorithms for quality/vulnerability resolutions and code coverage
- Use of models with greater contextual capacity
- Find pioneers to implement a solution in a real-world environment
- Integration with other tools
- Finetuning of the model for custom frameworks

Thank You

imola informatica

IMOLA · BOLOGNA · MILANO

www.imolainformatica.it | blog.imolainformatica.it