The Case for Explainability of Real-Time Systems and their Analyses

Bjorn Andersson

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

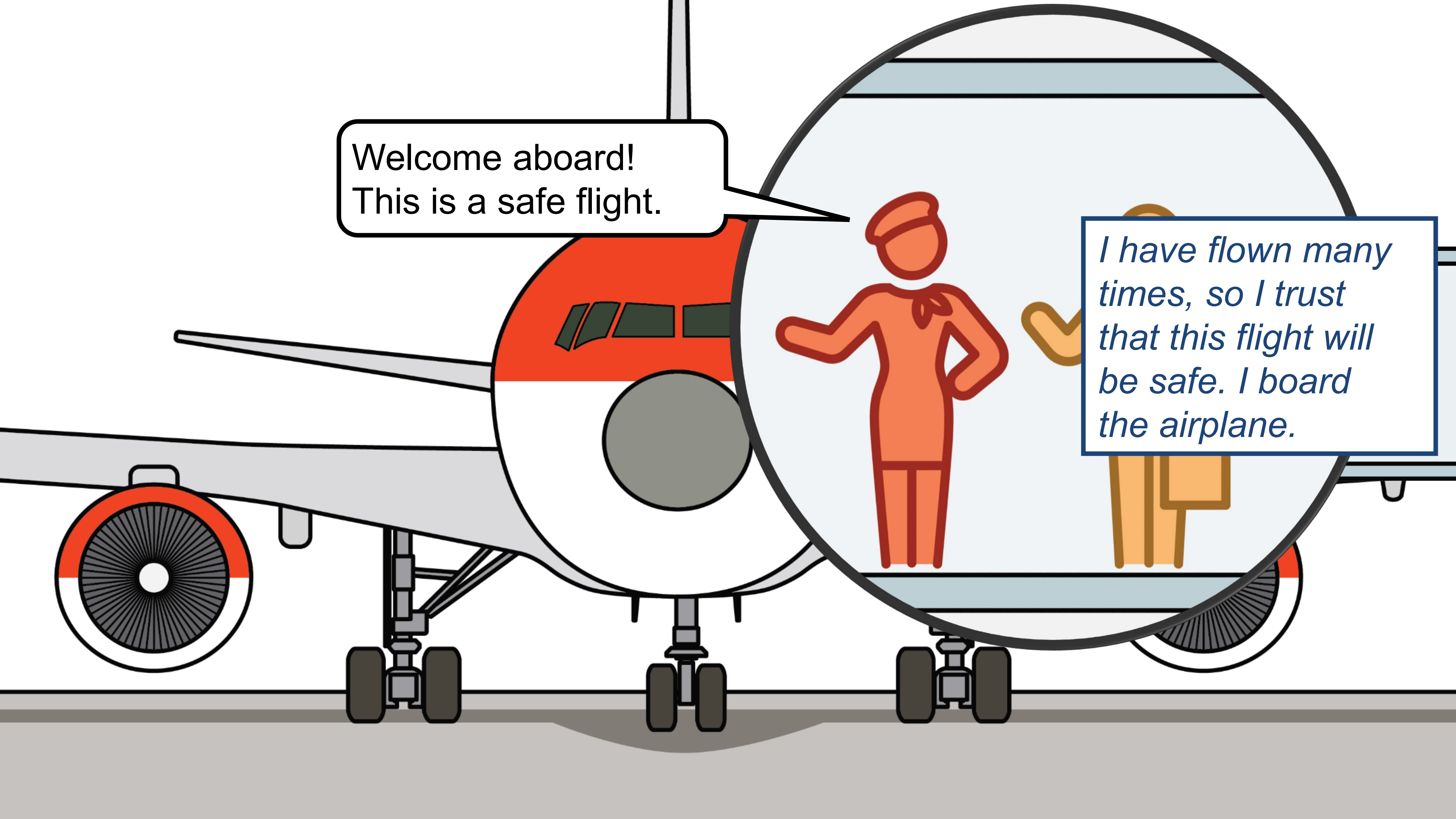
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

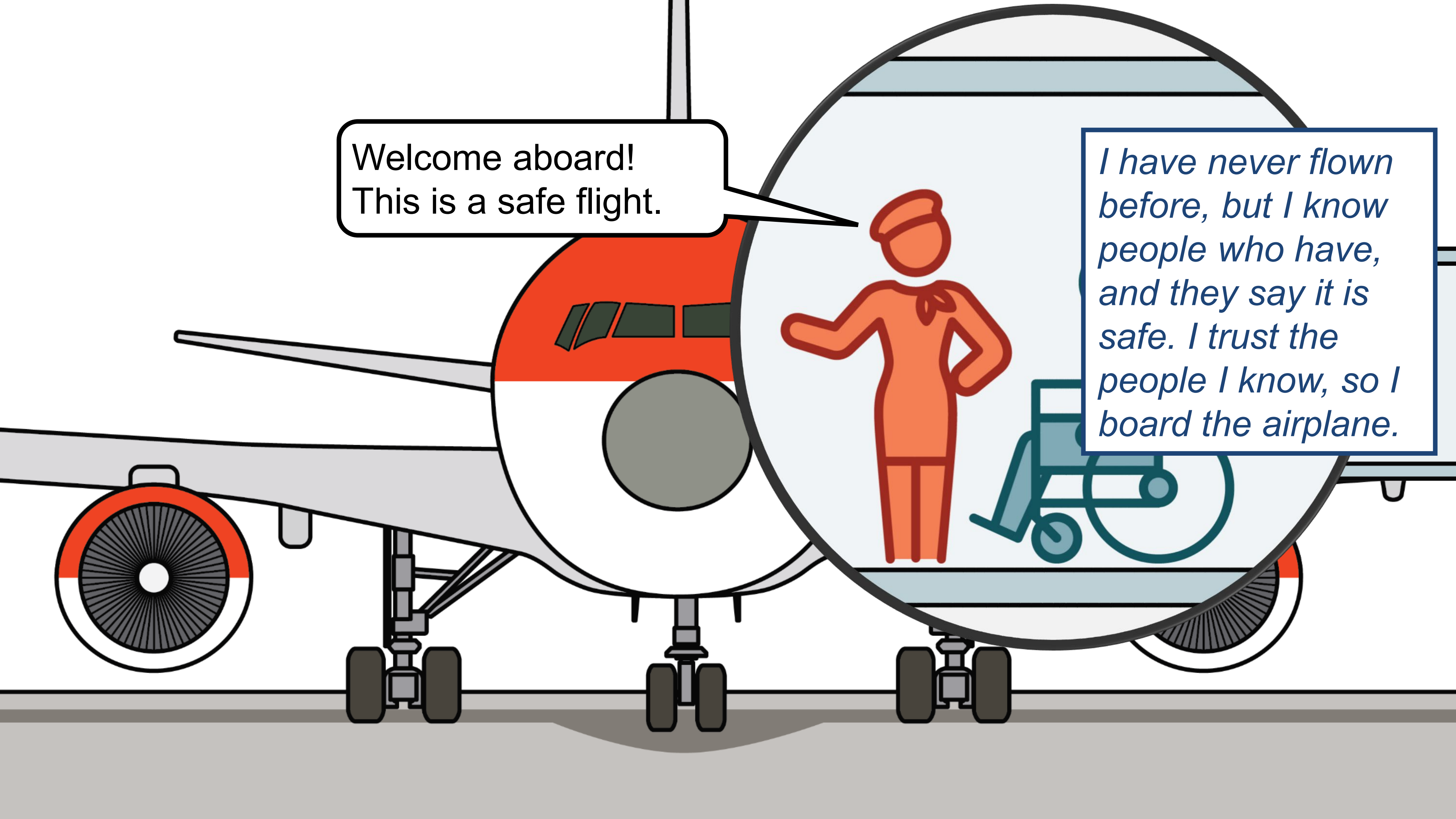
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM22-0896

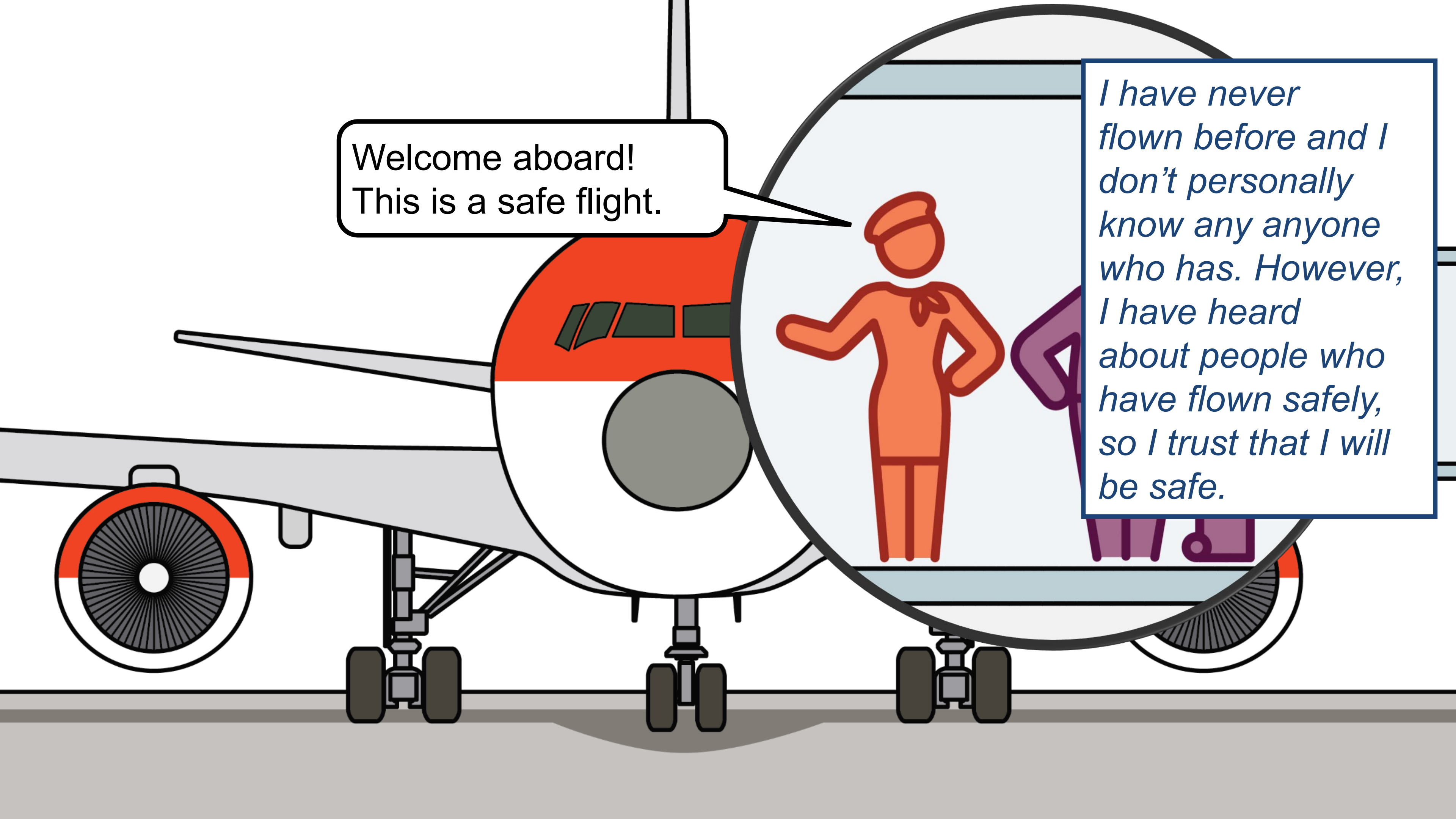
Motivation



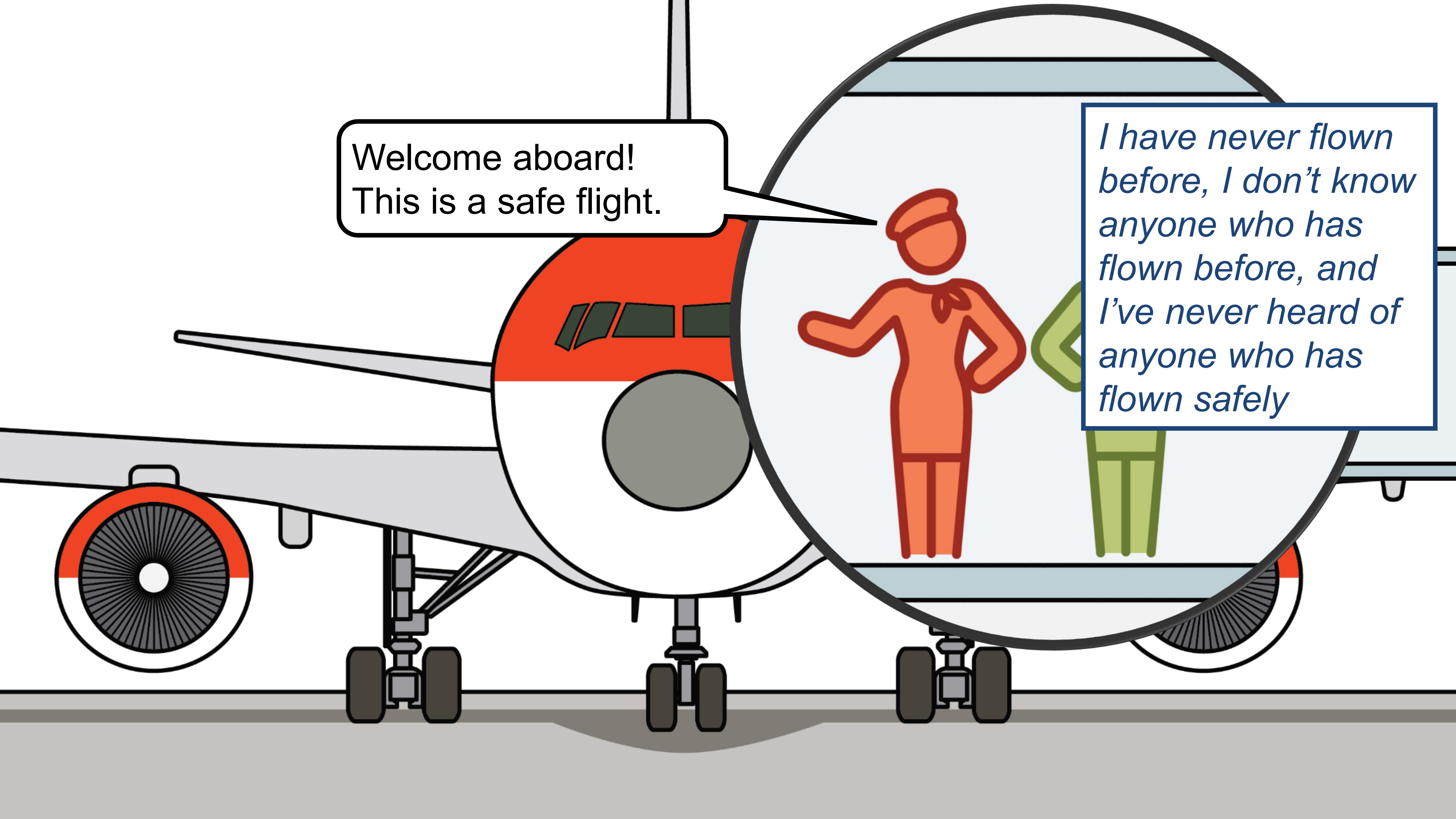
I have flown many times, so I trust that this flight will be safe. I board the airplane.



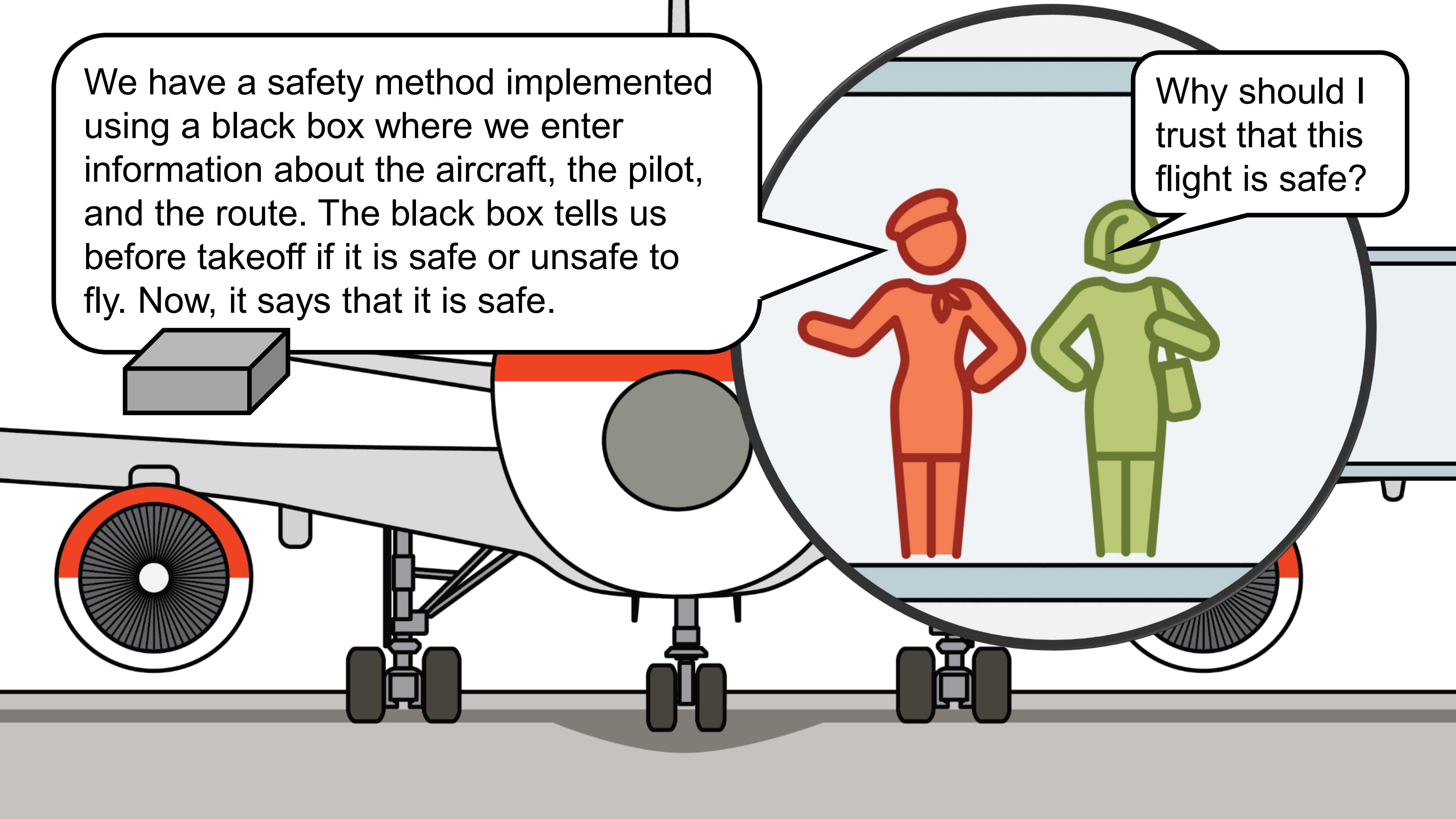
I have never flown before, but I know people who have, and they say it is safe. I trust the people I know, so I board the airplane.

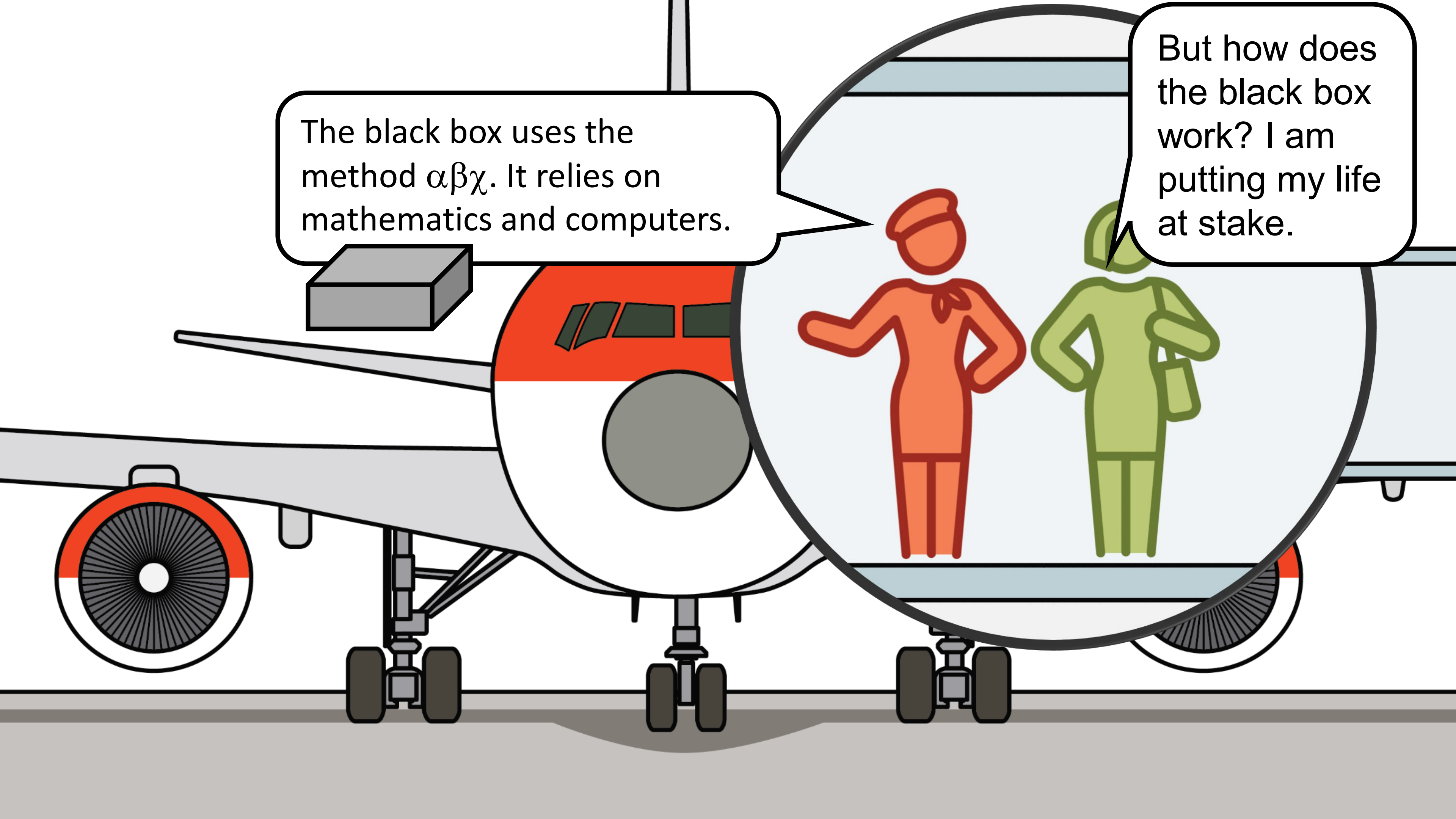


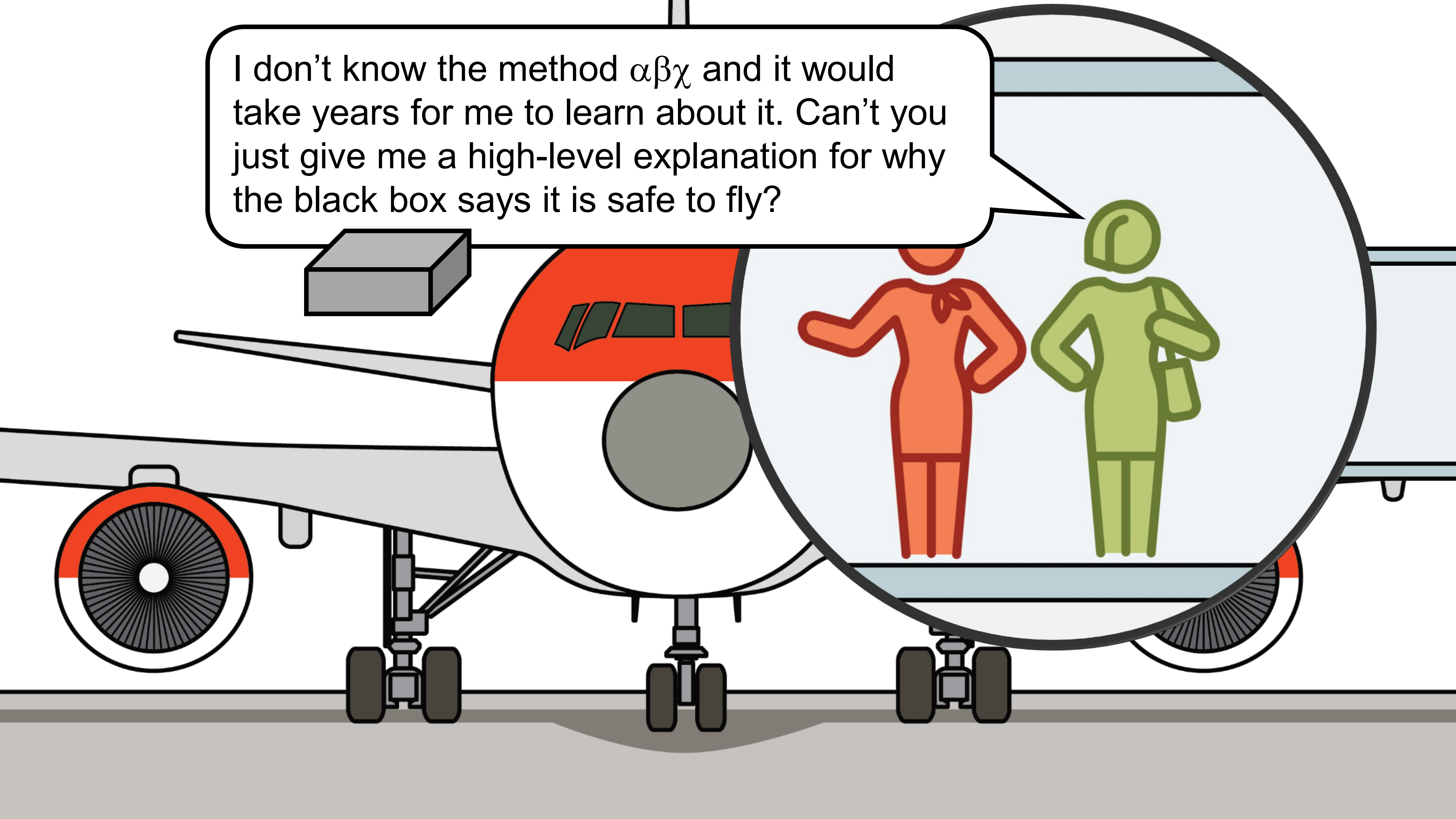
I have never flown before and I don’t personally know any anyone who has. However, I have heard about people who have flown safely, so I trust that I will be safe.

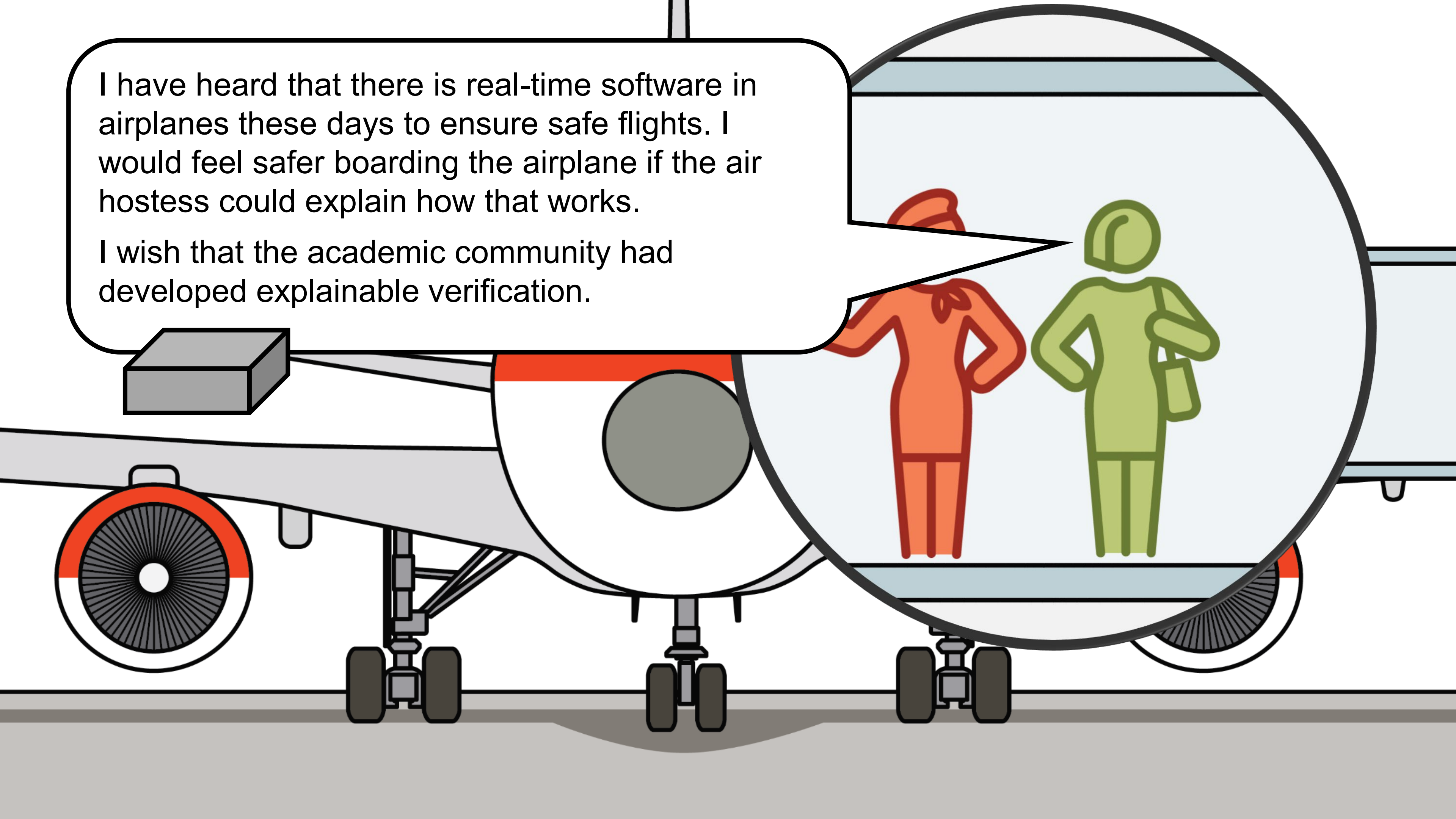


I have never flown before, I don't know anyone who has flown before, and I've never heard of anyone who has flown safely


We have a safety method implemented using a black box where we enter information about the aircraft, the pilot, and the route. The black box tells us before takeoff if it is safe or unsafe to fly. Now, it says that it is safe.
Why should I trust that this flight is safe?

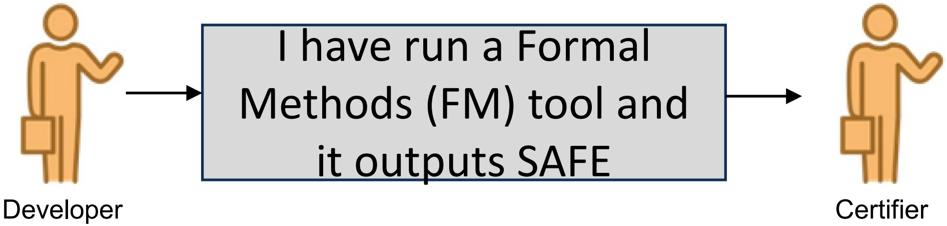

The black box uses the method αβχ. It relies on mathematics and computers.
But how does the black box work? I am putting my life at stake.


I don’t know the method αβχ and it would take years for me to learn about it. Can’t you just give me a high-level explanation for why the black box says it is safe to fly?


I have heard that there is real-time software in airplanes these days to ensure safe flights. I would feel safer boarding the airplane if the air hostess could explain how that works.

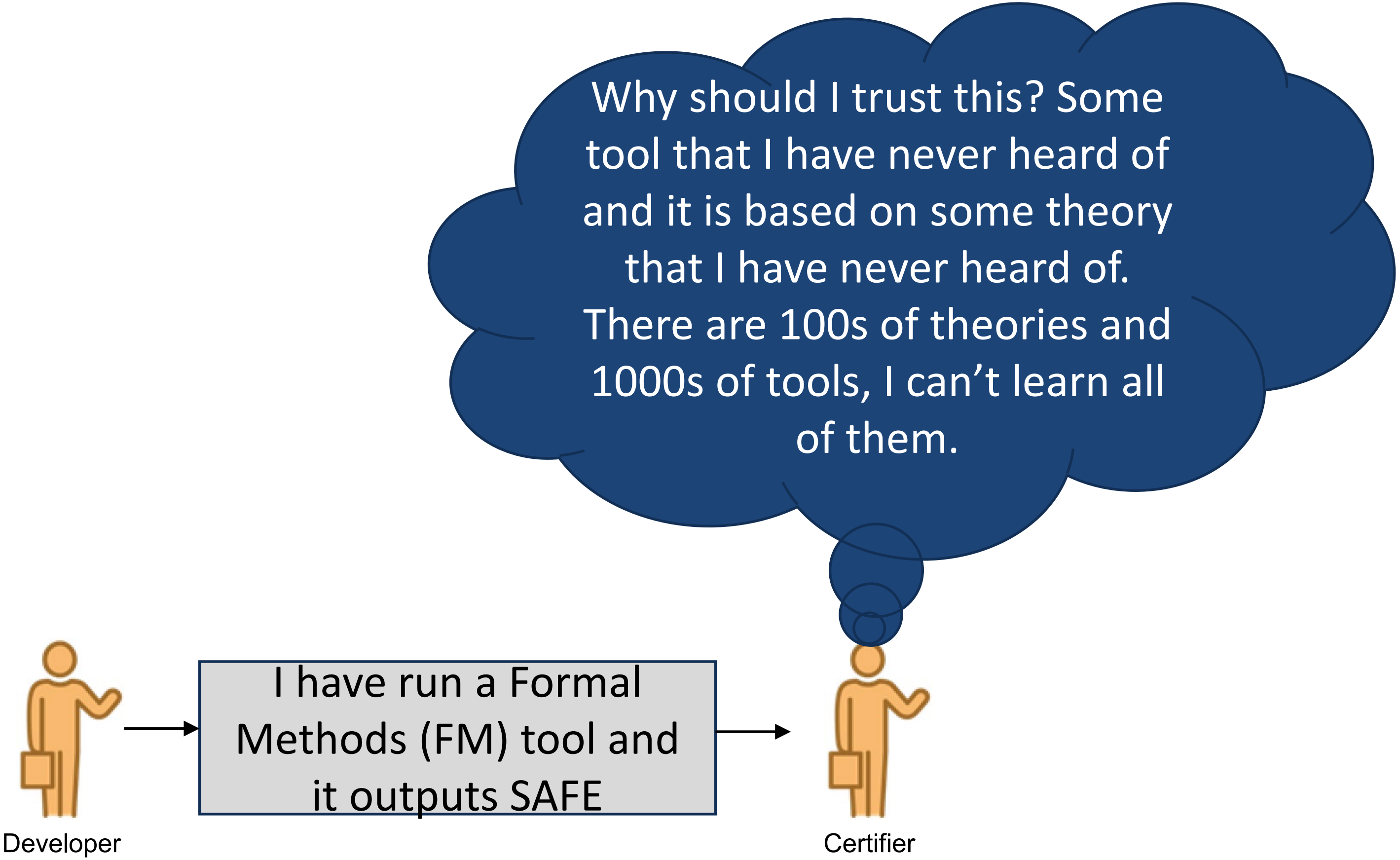
I wish that the academic community had developed explainable verification.

Problem: How to get developers and certifiers to trust FMs



I have run a Formal Methods (FM) tool and it outputs SAFE

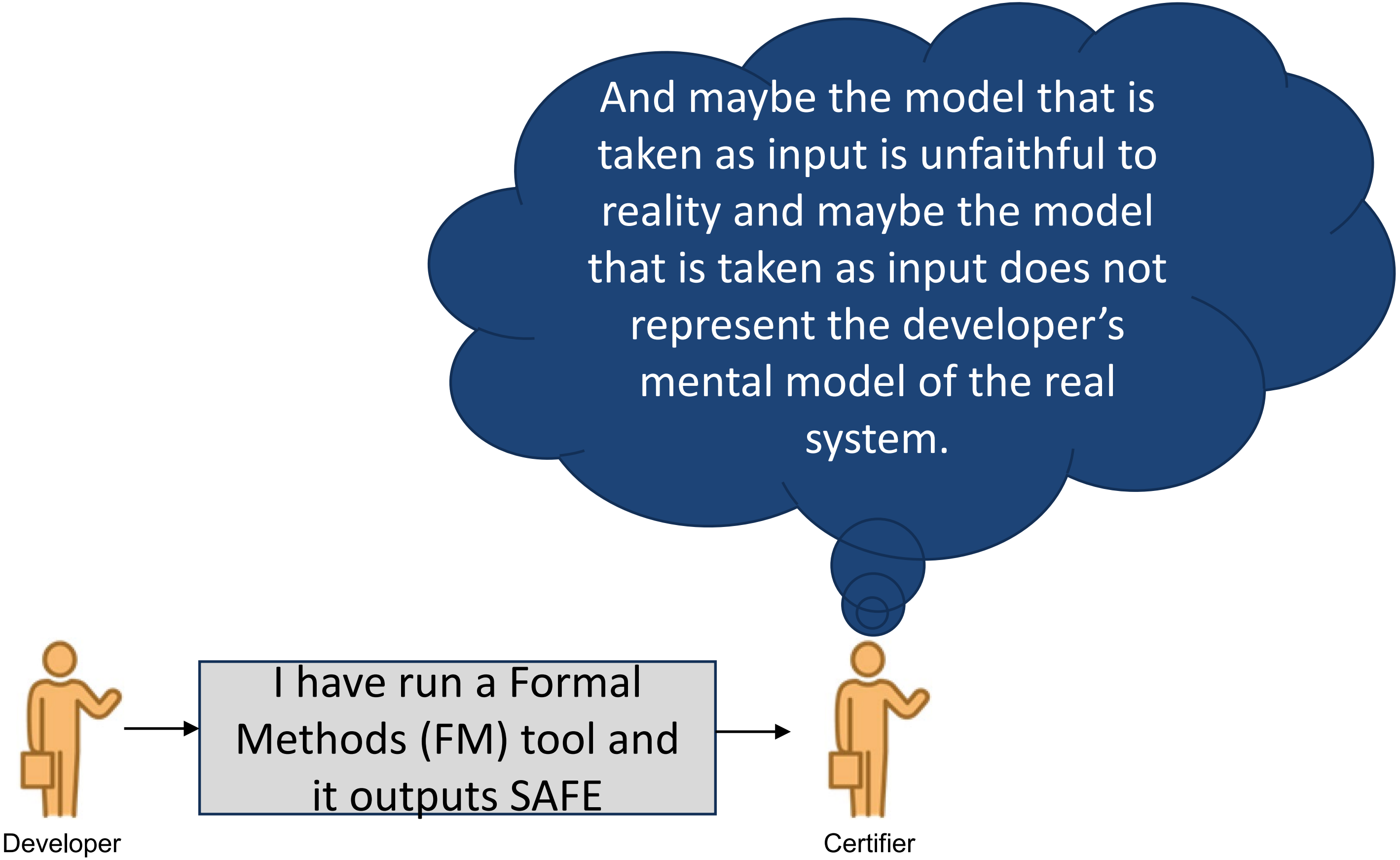
Problem: How to get developers and certifiers to trust FMs



Why should I trust this? Some tool that I have never heard of and it is based on some theory that I have never heard of. There are 100s of theories and 1000s of tools, I can't learn all of them.

I have run a Formal Methods (FM) tool and it outputs SAFE

Problem: How to get developers and certifiers to trust FMs



And maybe the model that is taken as input is unfaithful to reality and maybe the model that is taken as input does not represent the developer’s mental model of the real system.

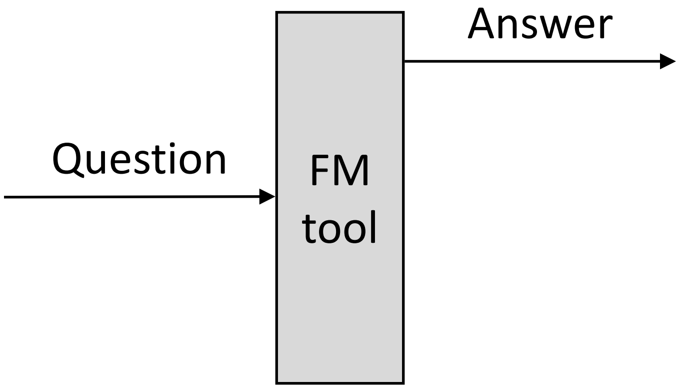
I have run a Formal Methods (FM) tool and it outputs SAFE

Developer

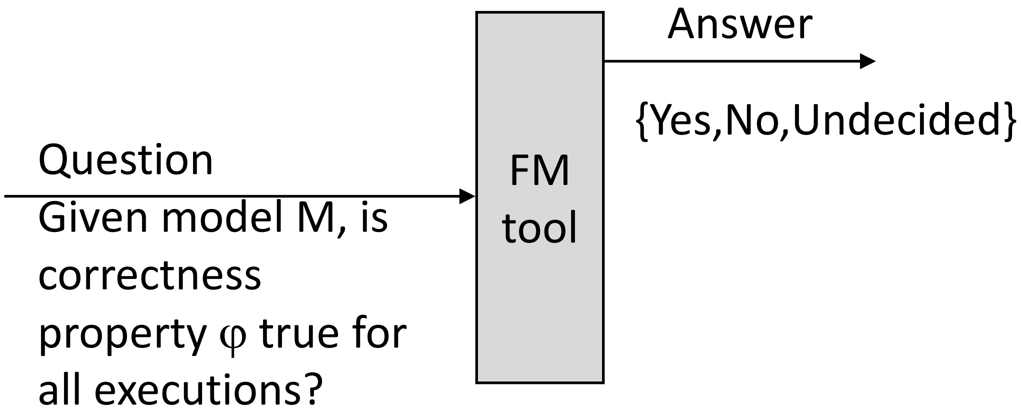
Certifier

Explanation of Analysis

The Present: A World Without Explainability



The Present: A World Without Explainability



Question

Given model M, is correctness property φ true for all executions?

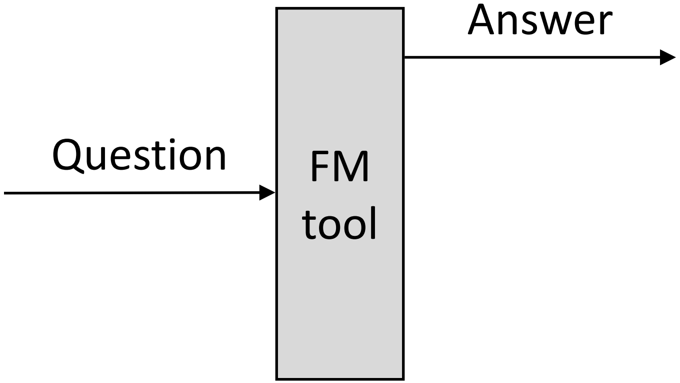
FM tool

Answer

{Yes,No,Undecided}

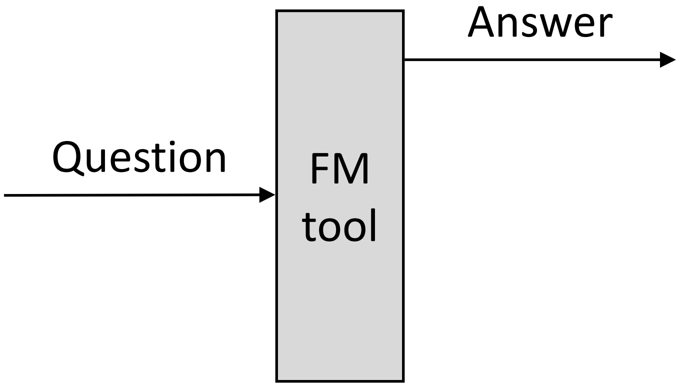
The Present: A World Without Explainability

Question: Given taskset τ, the following run-time scheduler S, and assumptions A, is it schedulable?



The Present: A World Without Explainability

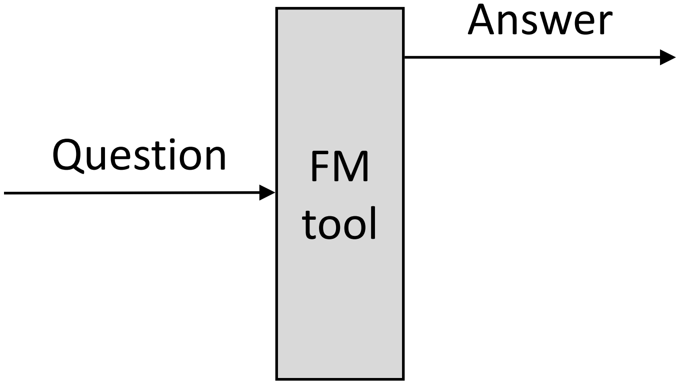
Question: Given taskset τ, the following run-time scheduler S, and assumptions A, is it schedulable?



Answer: Yes

The Present: A World Without Explainability

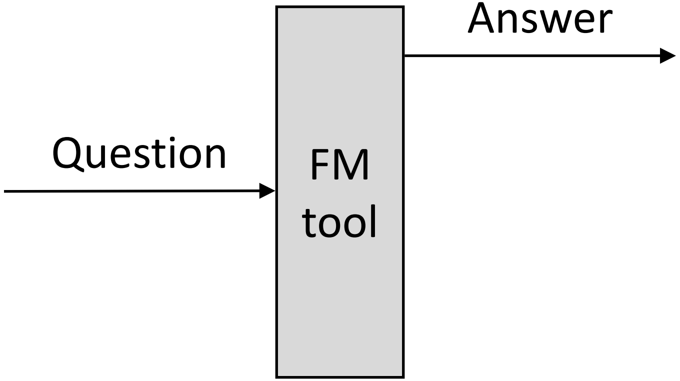
Question: Given taskset τ, the following run-time scheduler S, and assumptions A, is it unschedulable?



Answer: No

The Present: A World Without Explainability

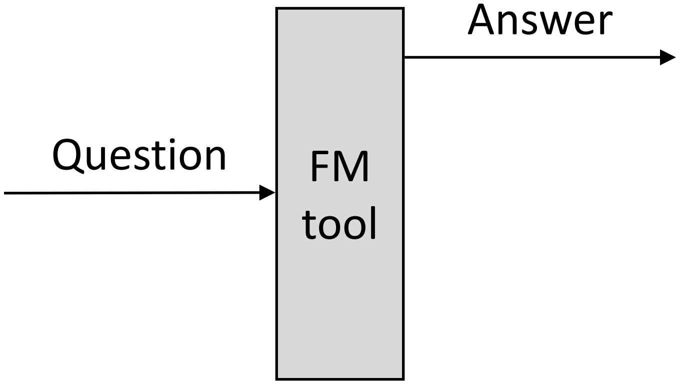
Question: Given program P and restriction R on inputs, what is its worst-case execution time?



Answer: 42 milliseconds

The Present: A World Without Explainability

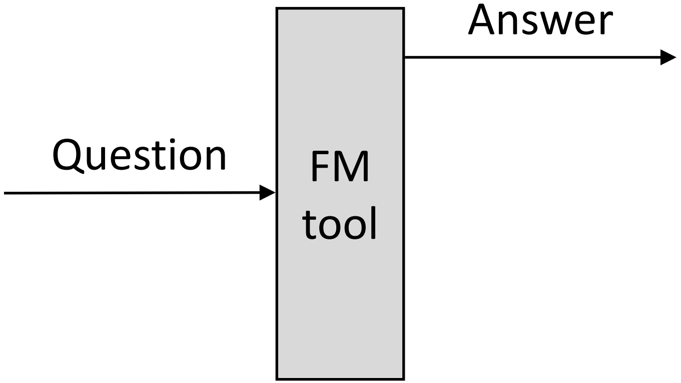
Question: Given program P and restriction R on inputs, does it hold for all executions that the execution time is less than or equal to TARGET_WCET_BOUND?



Answer: Yes

The Present: A World Without Explainability

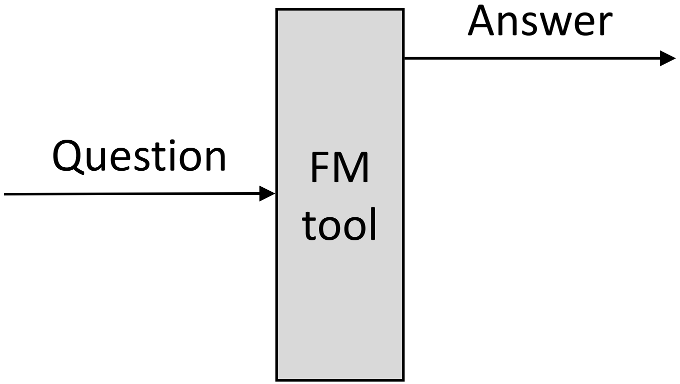
Question: Given program P, is assertion φ true for all executions?



Answer: No

The Present: A World Without Explainability

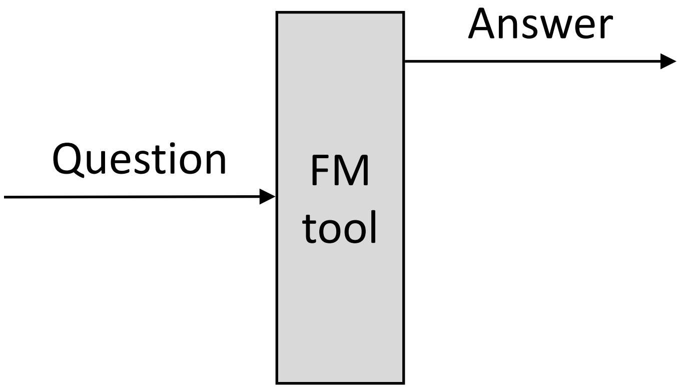
Question: Given program P, is assertion φ false for some execution?



Answer: Yes

The Present: A World Without Explainability

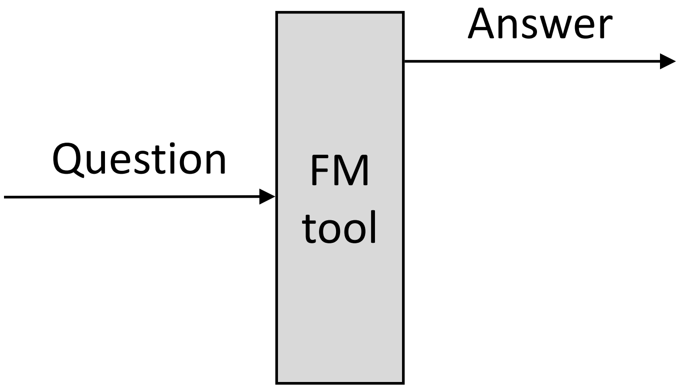
Question: Given the following executable code, is the assumption “no-task-suspension-can-occur-at-runtime” true?



Answer: Yes

The Present: A World Without Explainability

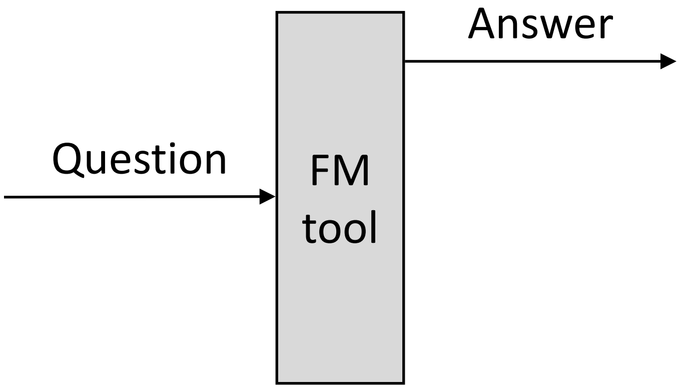
Question: Given the following plant and controller “x^{new} = A*x+B*u, u=K*x” does the schedule S of the task that executes the controller yield control performance J?



Answer: Yes

The Present: A World Without Explainability

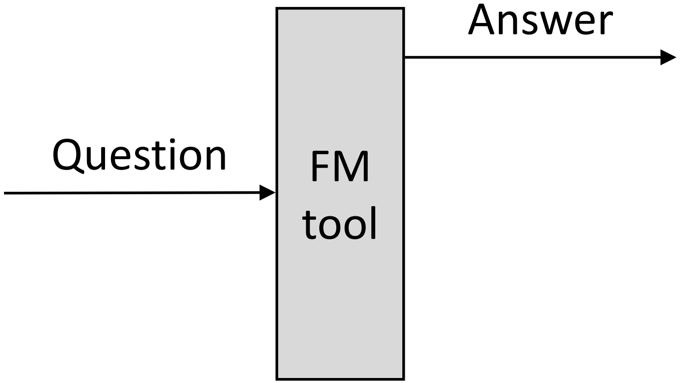
Question: Given the following plant and controller “x^{new} = A*x+B*u, u=K*x” does the set of schedules S of the task that executes the controller yield control performance J or better?



Answer: Yes

The Present: A World Without Explainability

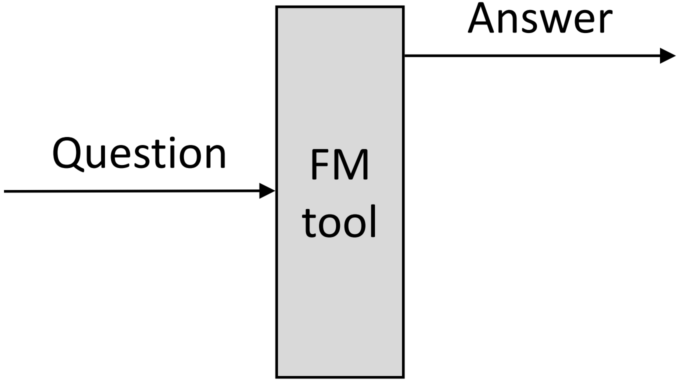
Question: Given the following human operator H, plant P, and control software C, does it hold that the mode perceived by H, P, and C differ by at most J?



Answer: Yes

The Present: A World Without Explainability

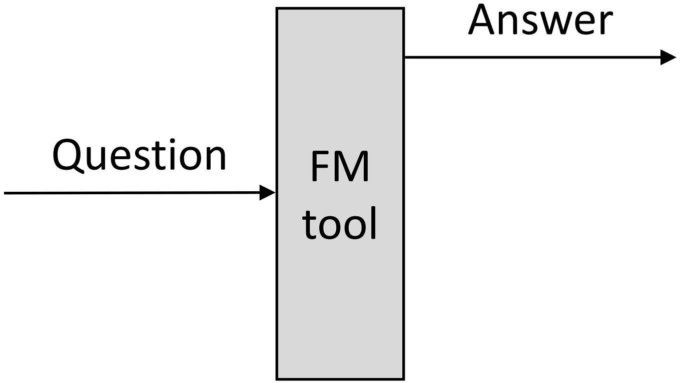
Question: Given the following distributed computer system D with message flow models S, and buffers B, does it hold that buffer overrun never occurs?



Answer: Yes

The Present: A World Without Explainability

Question: Given the following distributed computer system D with message flow models S, and buffers B, does it hold that buffer overrun never occurs?

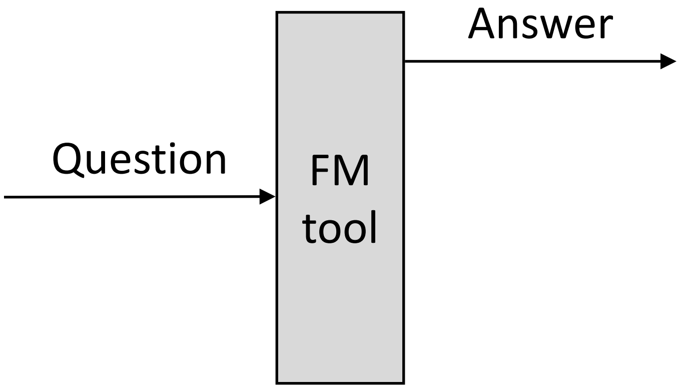


Answer: Yes

How much should one trust the FM tool that outputs answers?

The Present: A World Without Explainability

Question: Given the following distributed computer system D with message flow models S, and buffers B, does it hold that buffer overrun never occurs?



Question

FM tool

Answer

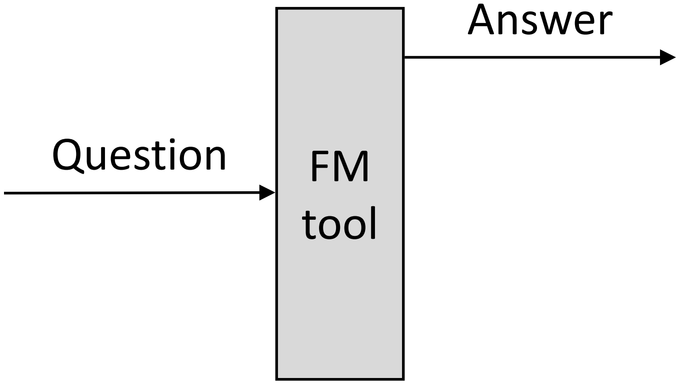
Answer: Yes

How much should one trust the FM tool that outputs answers?

If one does not know how the FM tool works, why should one trust it?

The Present: A World Without Explainability

Question: Given the following distributed computer system D with message flow models S, and buffers B, does it hold that buffer overrun never occurs?



Answer: Yes

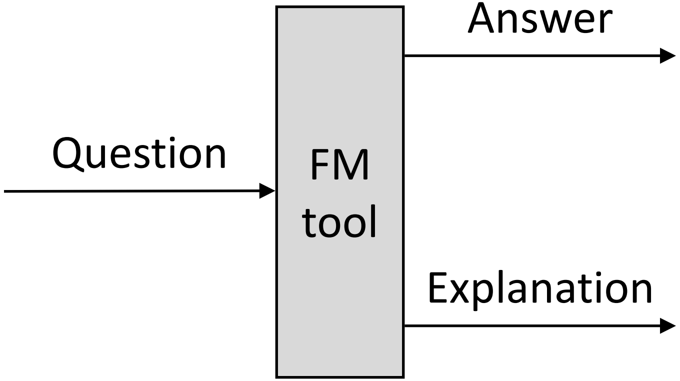
How much should one trust the FM tool that outputs answers?

If one does not know how the FM tool works, why should one trust it?

If practitioners do not trust the FM tools that researchers produce, how much value is there in the research?

A Potential Future: A World With Explainability

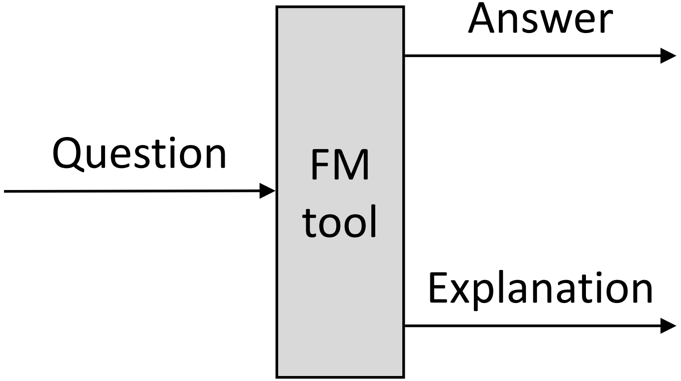
Question: Given taskset τ, the following run-time scheduler S, and assumptions A, is it schedulable?



Answer: Yes

A Potential Future: A World With Explainability

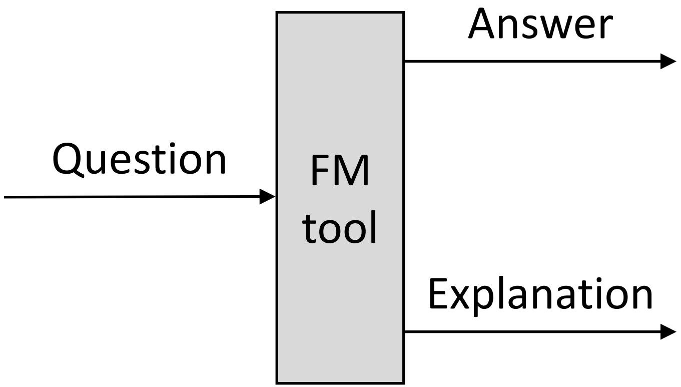
Question: Given program P and restriction R on inputs, what is its worst-case execution time?



Answer: 42 milliseconds

A Potential Future: A World With Explainability

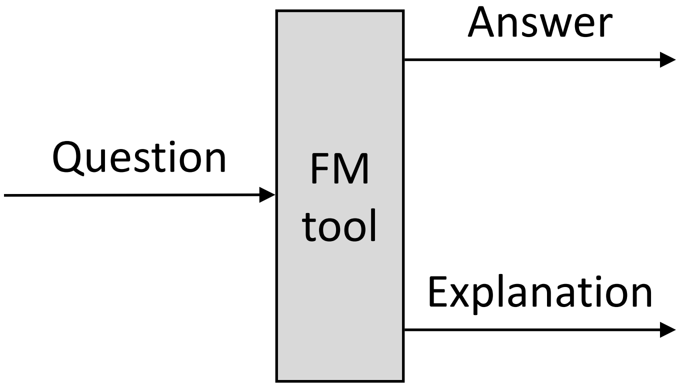
Question: Given program P and restriction R on inputs, does it hold for all executions that the execution time is less than or equal to TARGET_WCET_BOUND?



Answer: Yes

A Potential Future: A World With Explainability

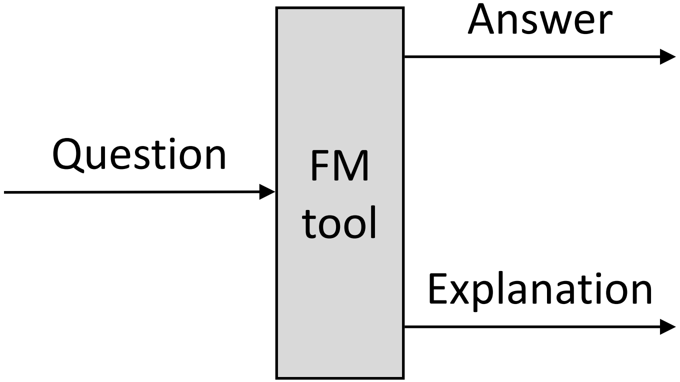
Question: Given program P, is assertion φ true for all executions?



Answer: Yes

A Potential Future: A World With Explainability

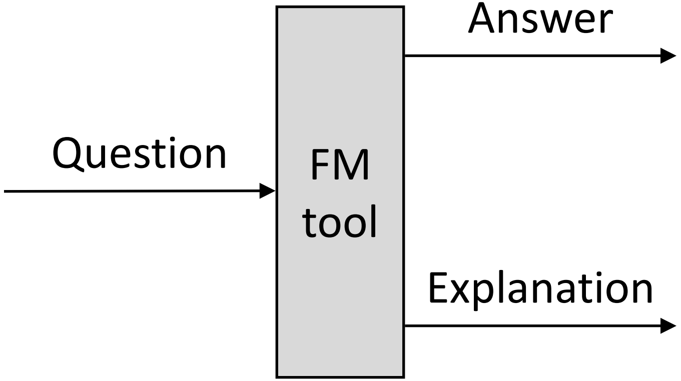
Question: Given program P, is assertion φ false for some execution?



Answer: No

A Potential Future: A World With Explainability

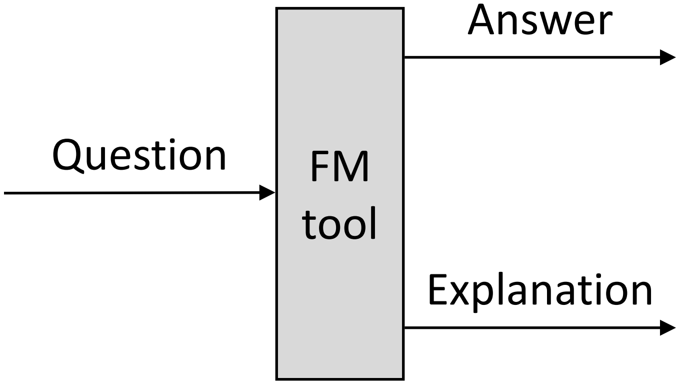
Question: Given the following taskset, is the assumption “no-task-suspension-can-occur-at-runtime” true?



Answer: Yes

A Potential Future: A World With Explainability

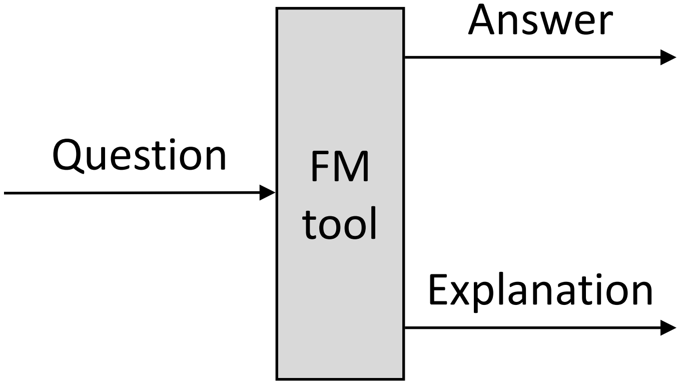
Question: Given the following plant and controller “x^{new} = A*x+B*u, u=K*x” does the schedule S of the task that executes the controller yield control performance J?



Answer: Yes

A Potential Future: A World With Explainability

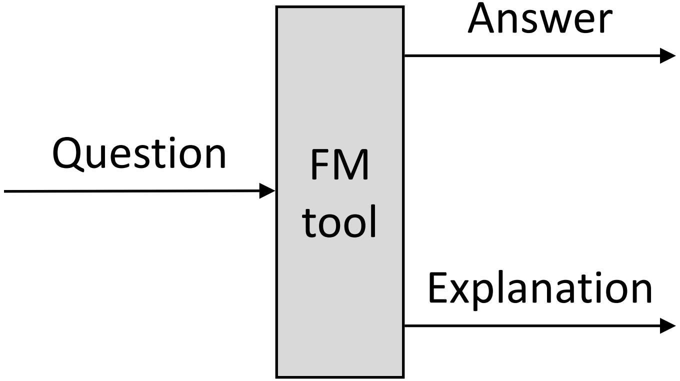
Question: Given the following plant and controller “x^{new} = A*x+B*u, u=K*x” does the set of schedules S of the task that executes the controller yield control performance J or better?



Answer: Yes

A Potential Future: A World With Explainability

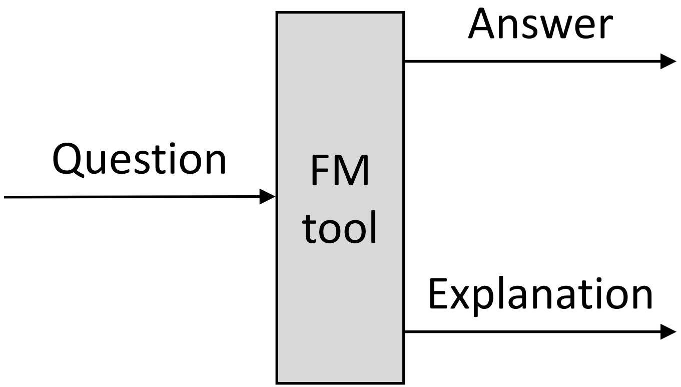
Question: Given the following human operator H, plant P, and control software C, does it hold that the mode perceived by H, P, and C differ by at most J?



Answer: Yes

A Potential Future: A World With Explainability

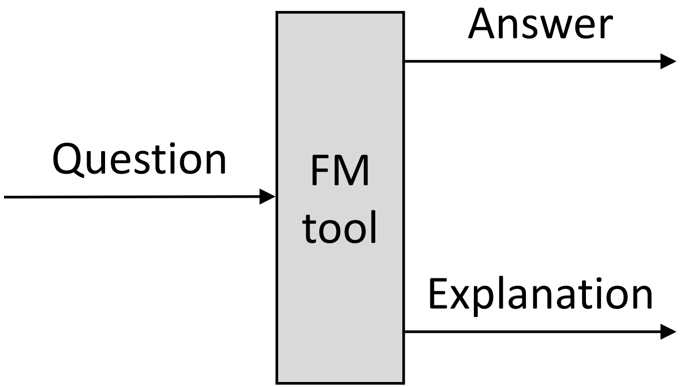
Question: Given the following distributed computer system D with message flow models S, and buffers B, does it hold that buffer overrun never occurs?



Answer: Yes

A Potential Future: A World With Explainability

Question: Given the following distributed computer system D with message flow models S, and buffers B, does it hold that buffer overrun never occurs?



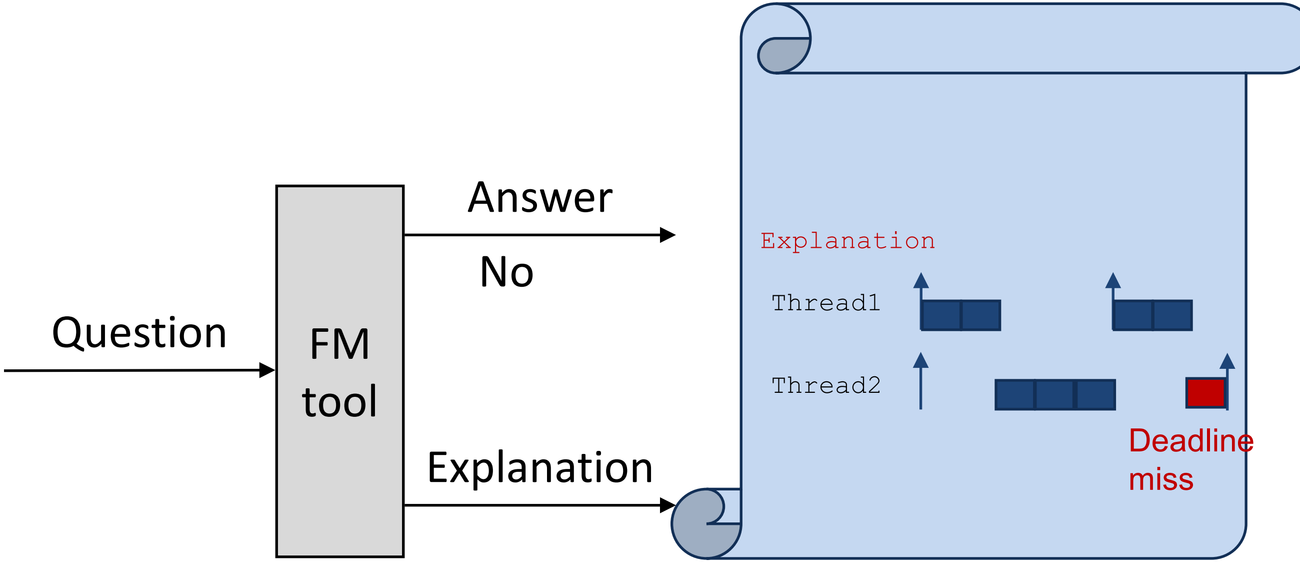
Answer: Yes

Explanability has the potential to make practitioners trust FM tools

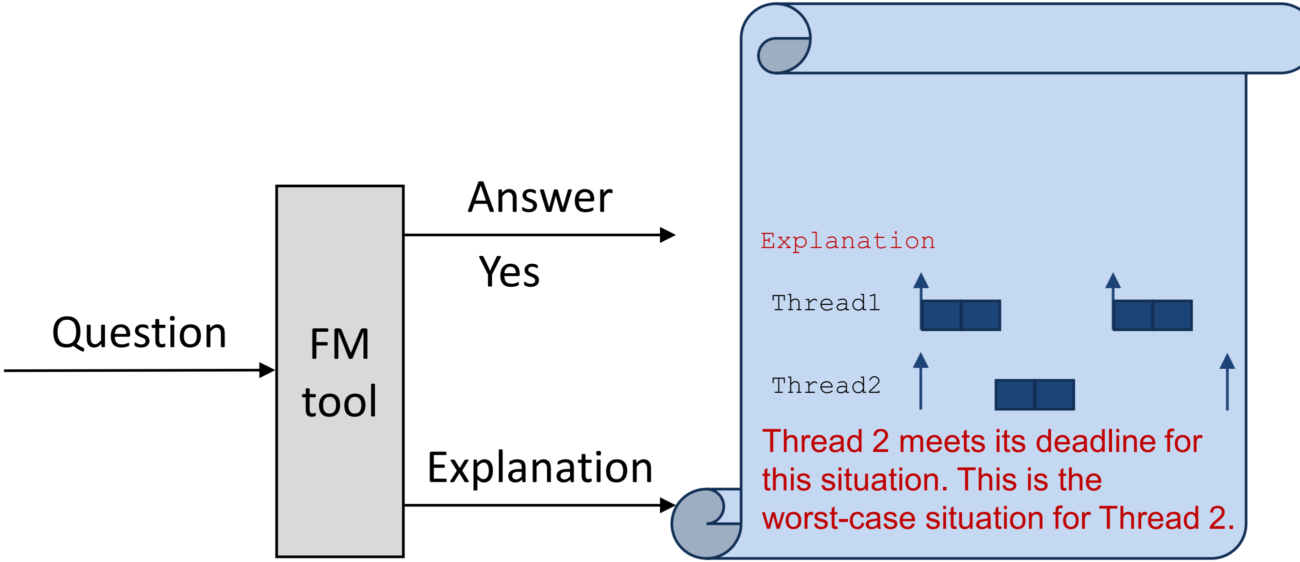
Explanability has the potential to make sure our research becomes more valuable to practitioners

What Could an Explanation Look Like?

Question: Given taskset τ, the following run-time scheduler S, and assumptions A, is it schedulable?



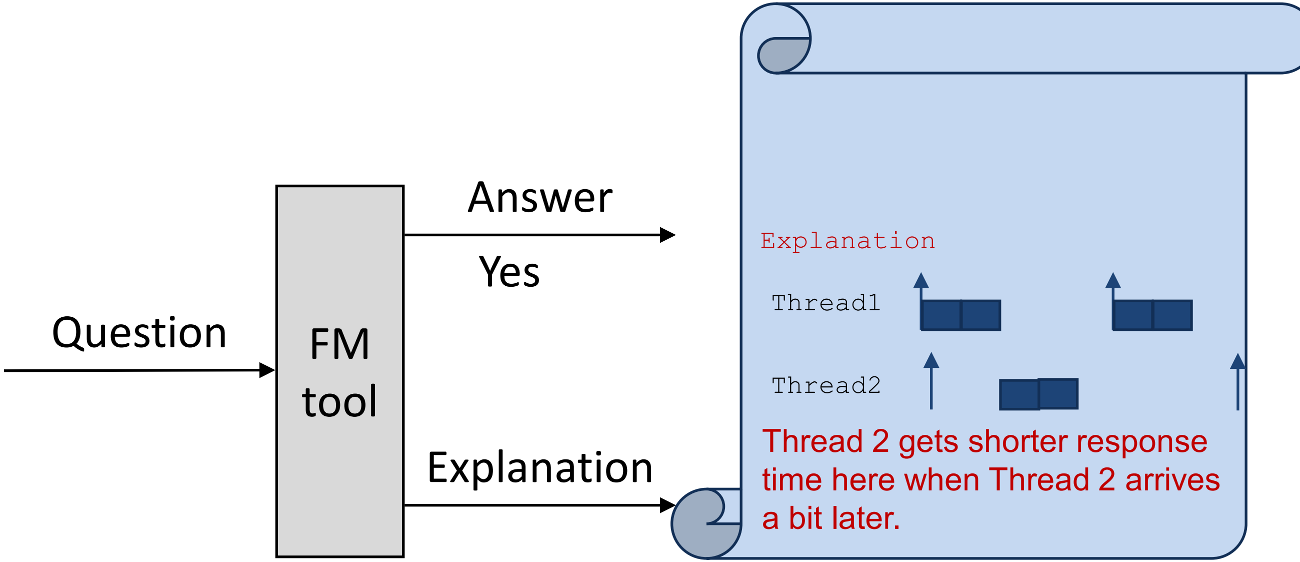
Question: Given taskset τ‘, the following run-time scheduler S, and assumptions A, is it schedulable?



Thread 2 meets its deadline for this situation. This is the worst-case situation for Thread 2.

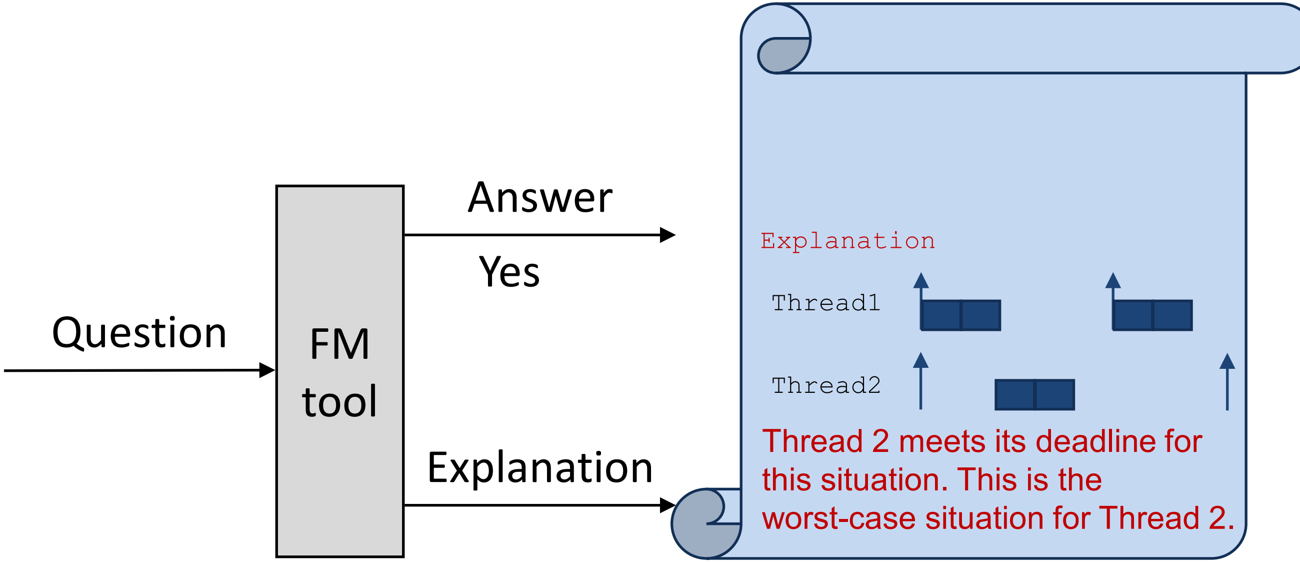
Click here to understand why this is the worst-case situation

Question: Given taskset τ‘, the following run-time scheduler S, and assumptions A, is it schedulable?



Thread 2 gets shorter response time here when Thread 2 arrives a bit later.

Question: Given taskset τ‘, the following run-time scheduler S, and assumptions A, is it schedulable?



Question

FM tool

Answer

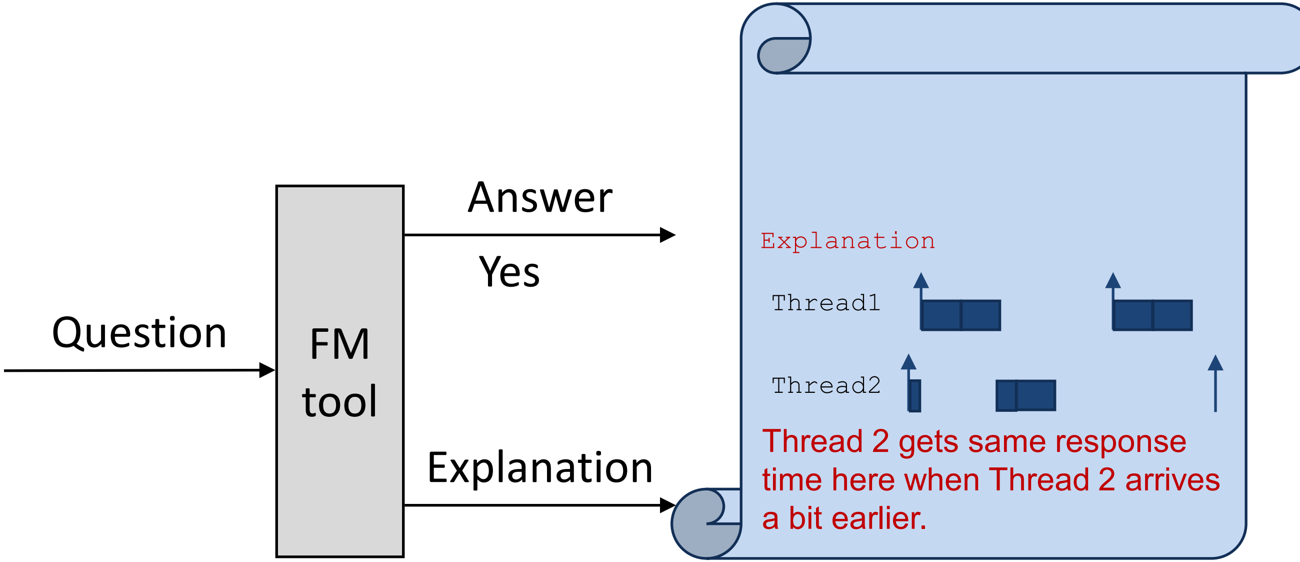
Yes

Explanation

Explanation

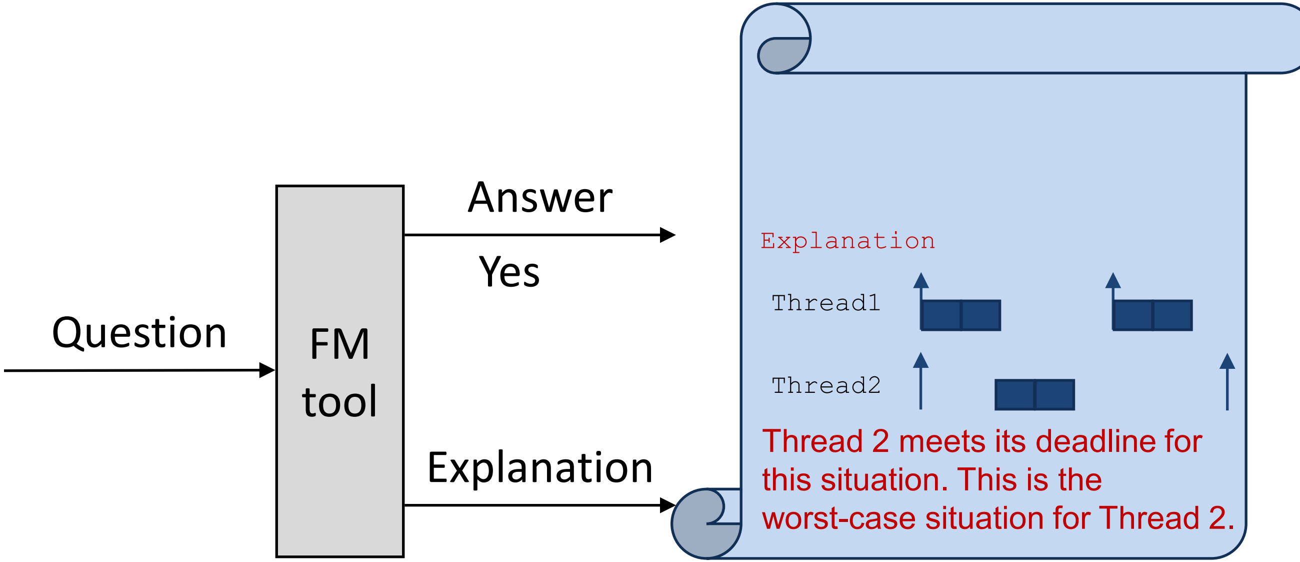
Thread 2 meets its deadline for this situation. This is the worst-case situation for Thread 2.

Question: Given taskset τ‘, the following run-time scheduler S, and assumptions A, is it schedulable?



Thread 2 gets same response time here when Thread 2 arrives a bit earlier.

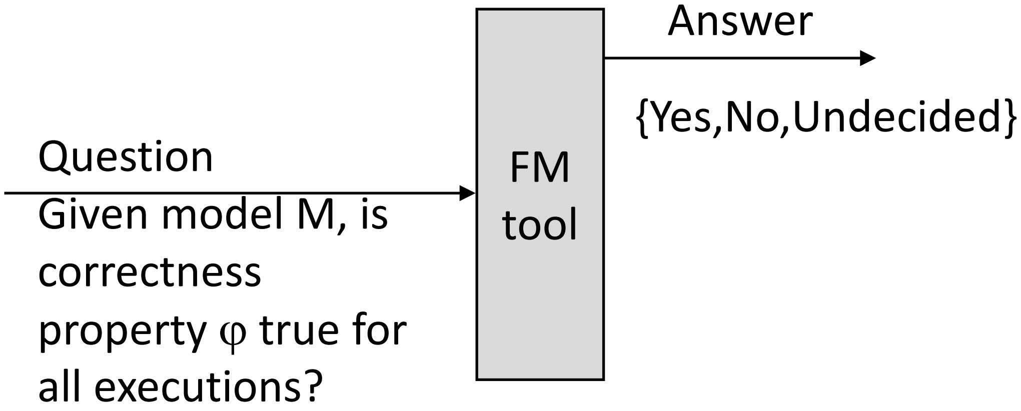
Question: Given taskset τ‘, the following run-time scheduler S, and assumptions A, is it schedulable?



Thread 2 meets its deadline for this situation. This is the worst-case situation for Thread 2.

Explanation of Input

Explainability of Input



Question

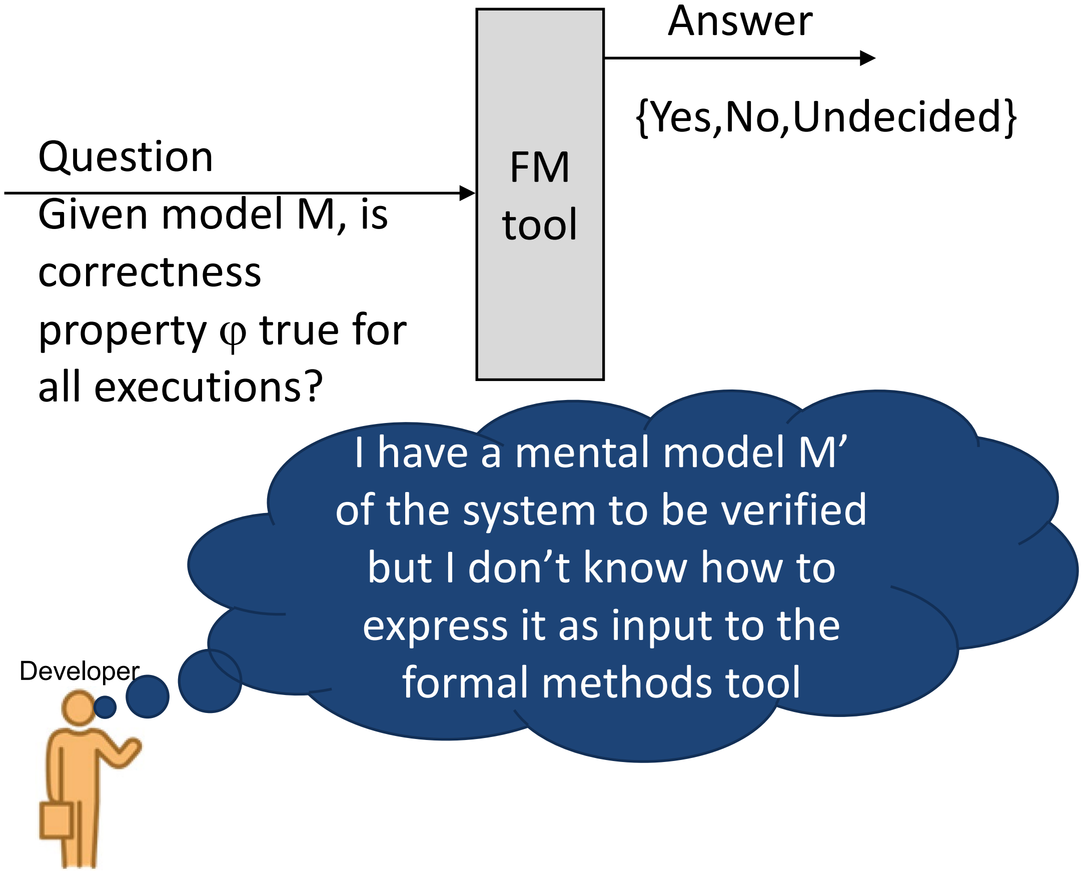
Given model M, is correctness property φ true for all executions?

FM tool

Answer

{Yes,No,Undecided}

Explainability of Input



Answer

{Yes,No,Undecided}

Question

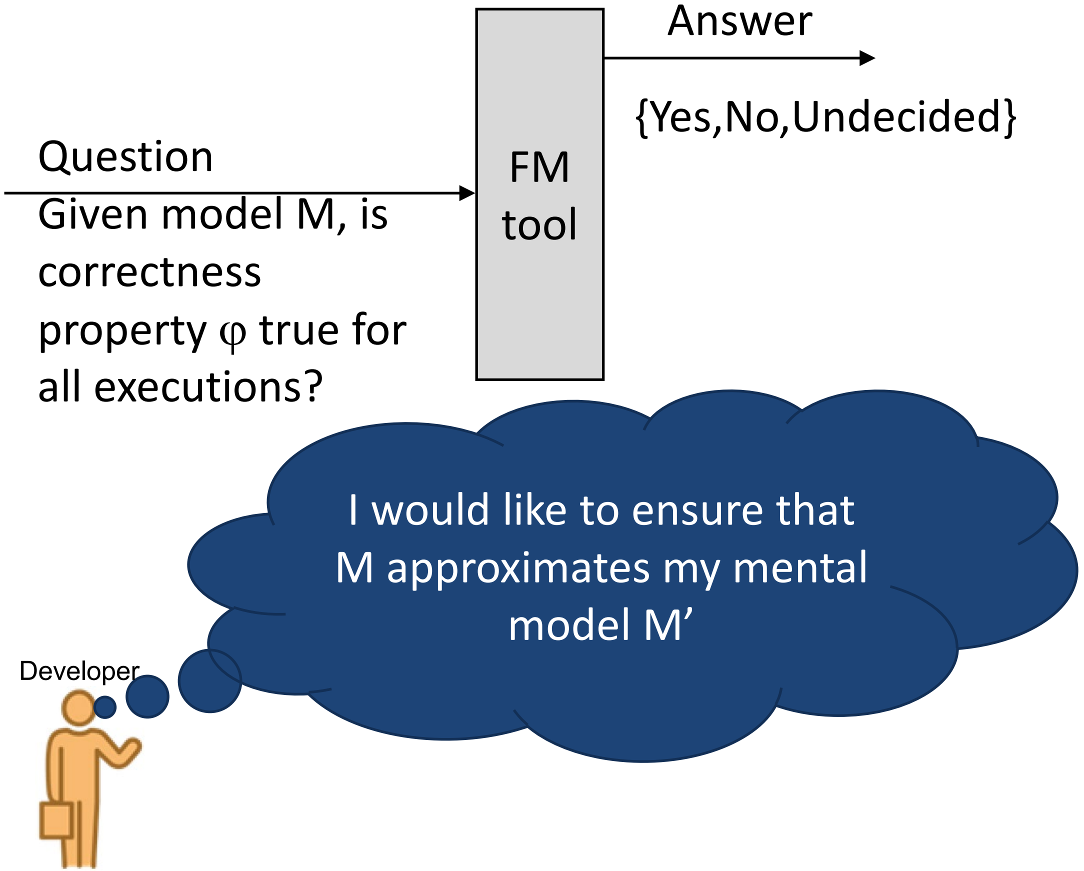
Given model M, is correctness property φ true for all executions?

FM tool

I have a mental model M’ of the system to be verified but I don’t know how to express it as input to the formal methods tool

Developer

Explainability of Input



Answer

{Yes,No,Undecided}

Question

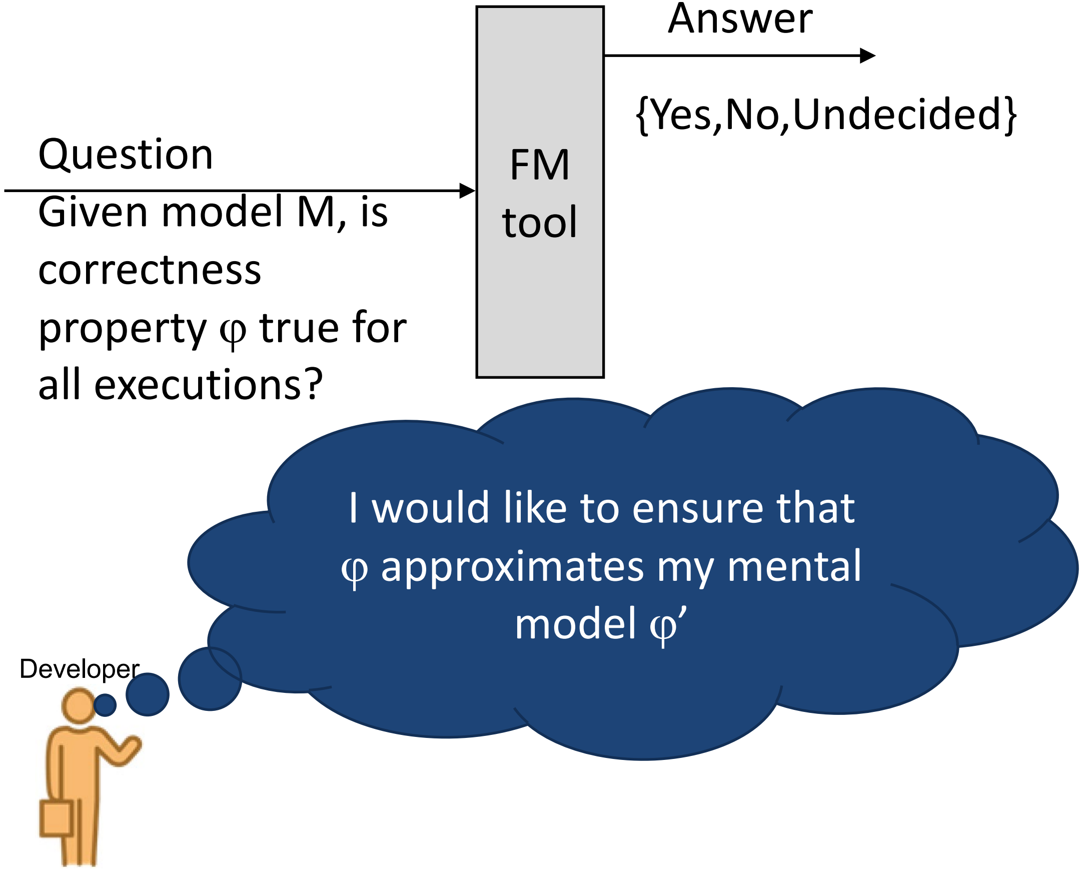
Given model M, is correctness property φ true for all executions?

FM tool

I would like to ensure that M approximates my mental model M'

Developer

Explainability of Input



Question

Given model M, is correctness property φ true for all executions?

FM tool

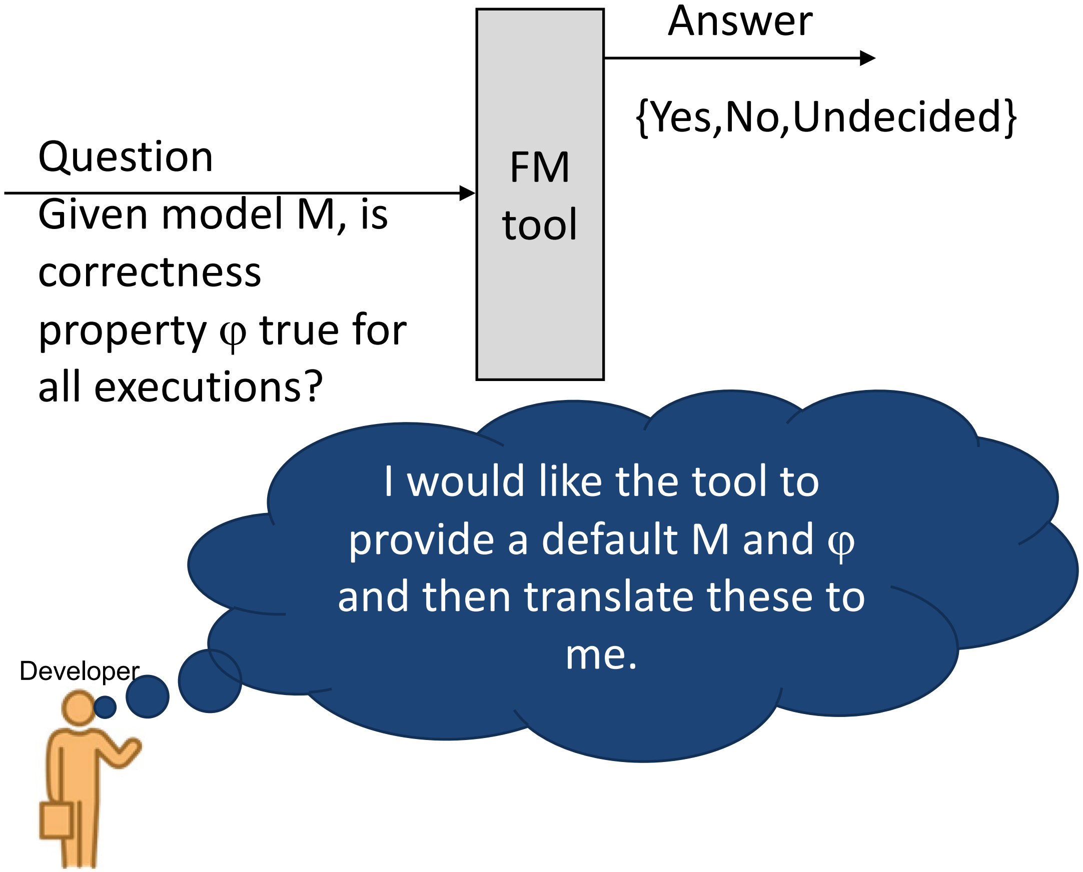
Answer

{Yes,No,Undecided}

Developer

I would like to ensure that φ approximates my mental model φ’

Explainability of Input



Question

Given model M, is correctness property φ true for all executions?

FM tool

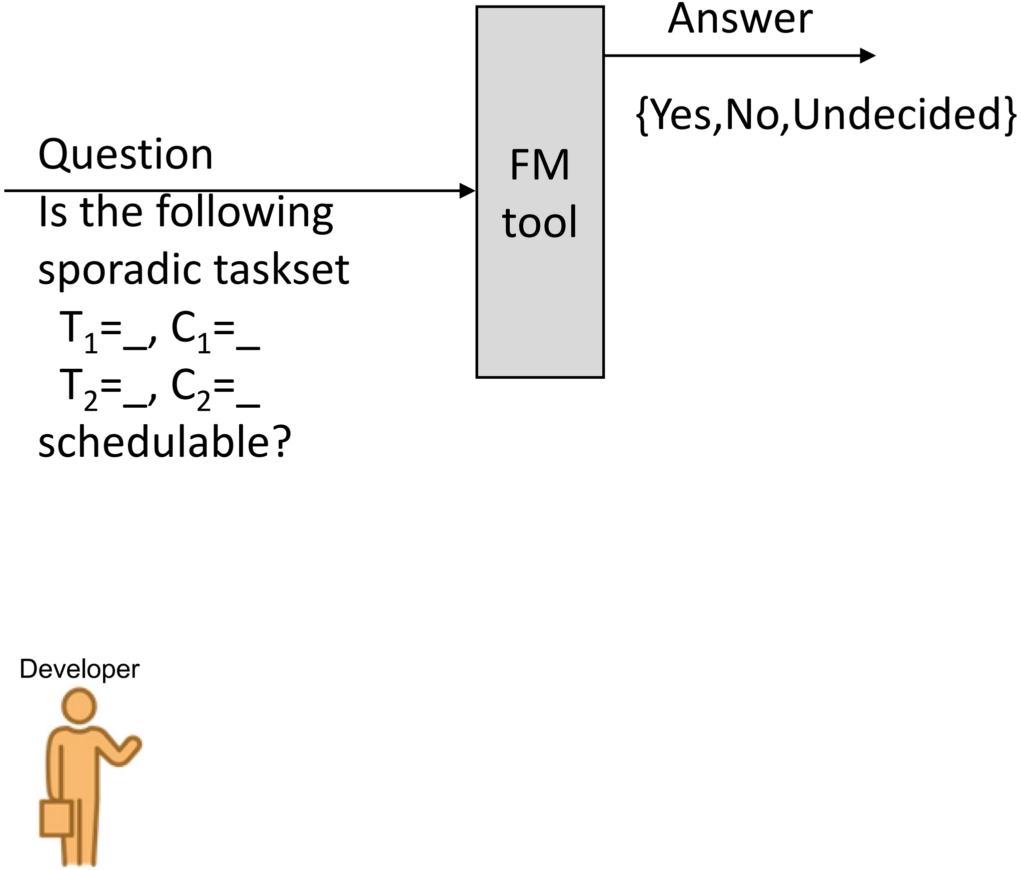
Answer

{Yes,No,Undecided}

I would like the tool to provide a default M and φ and then translate these to me.

Developer

Explainability of Input: Real-Time Scheduling



Question

Is the following sporadic taskset

T_1=_, C_1=_

T_2=_, C_2=_

schedulable?

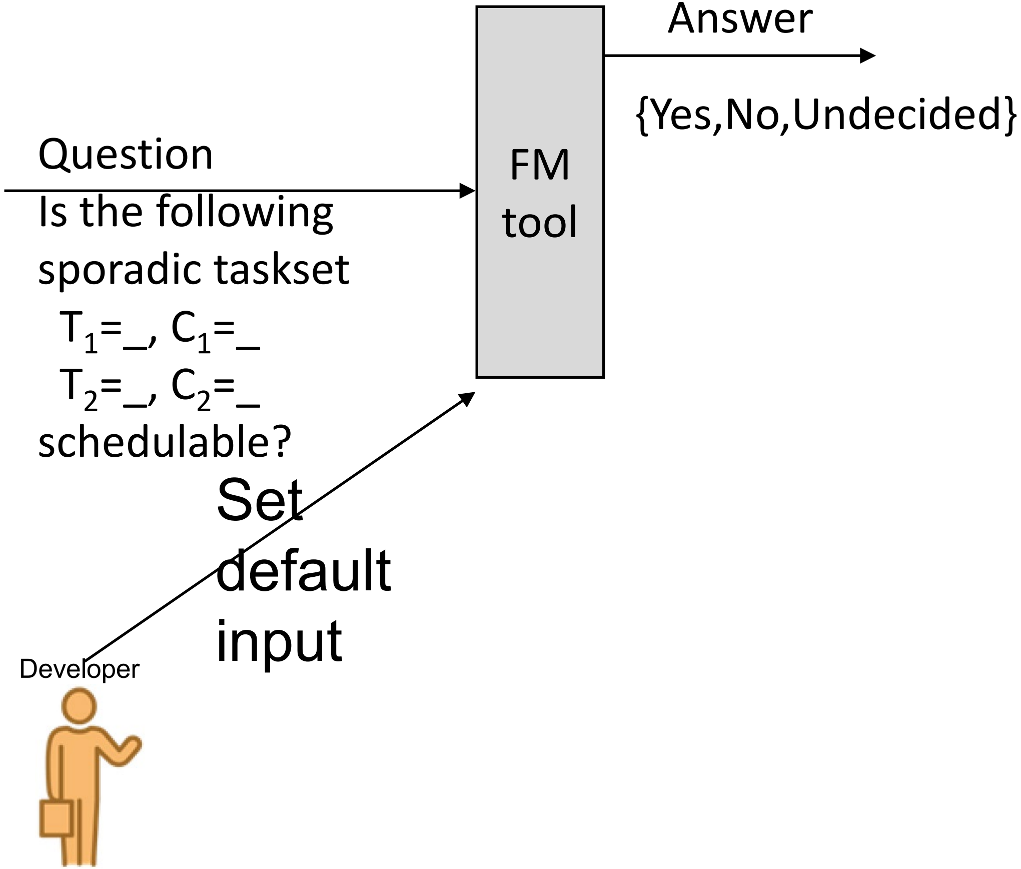
FM tool

Answer

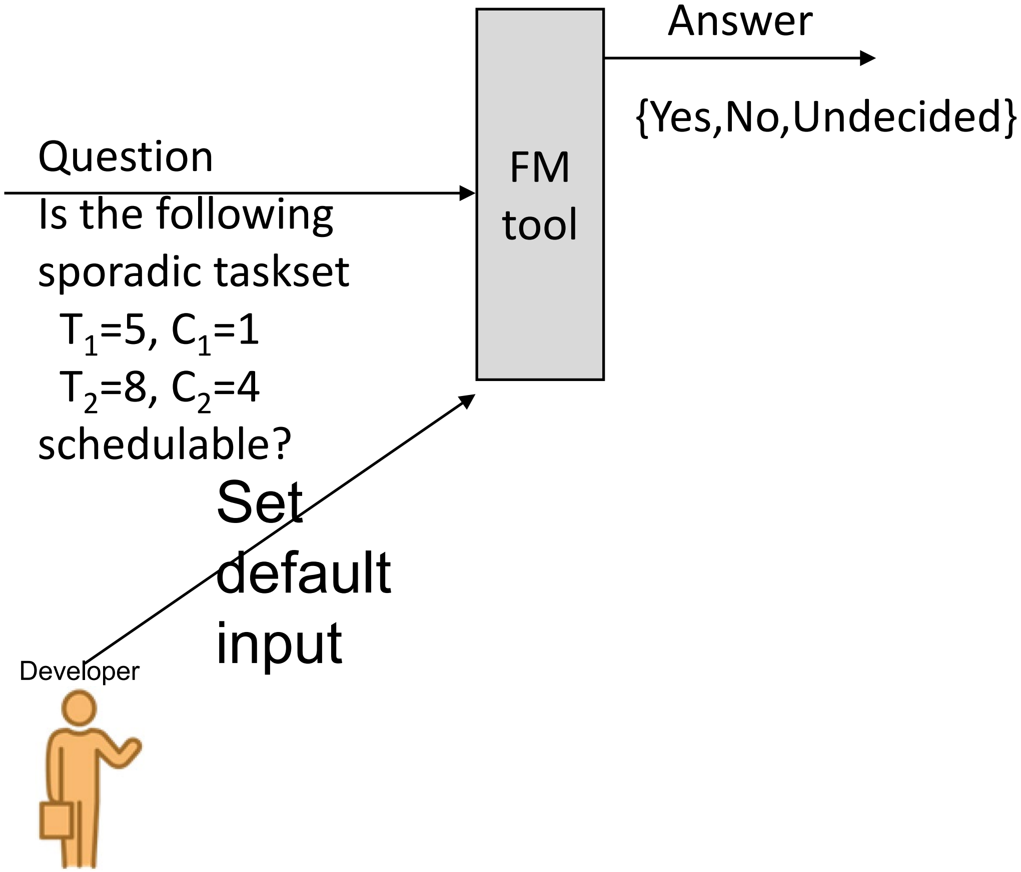
{Yes,No,Undecided}

Developer

Explainability of Input: Real-Time Scheduling



Explainability of Input: Real-Time Scheduling



Question

Is the following sporadic taskset

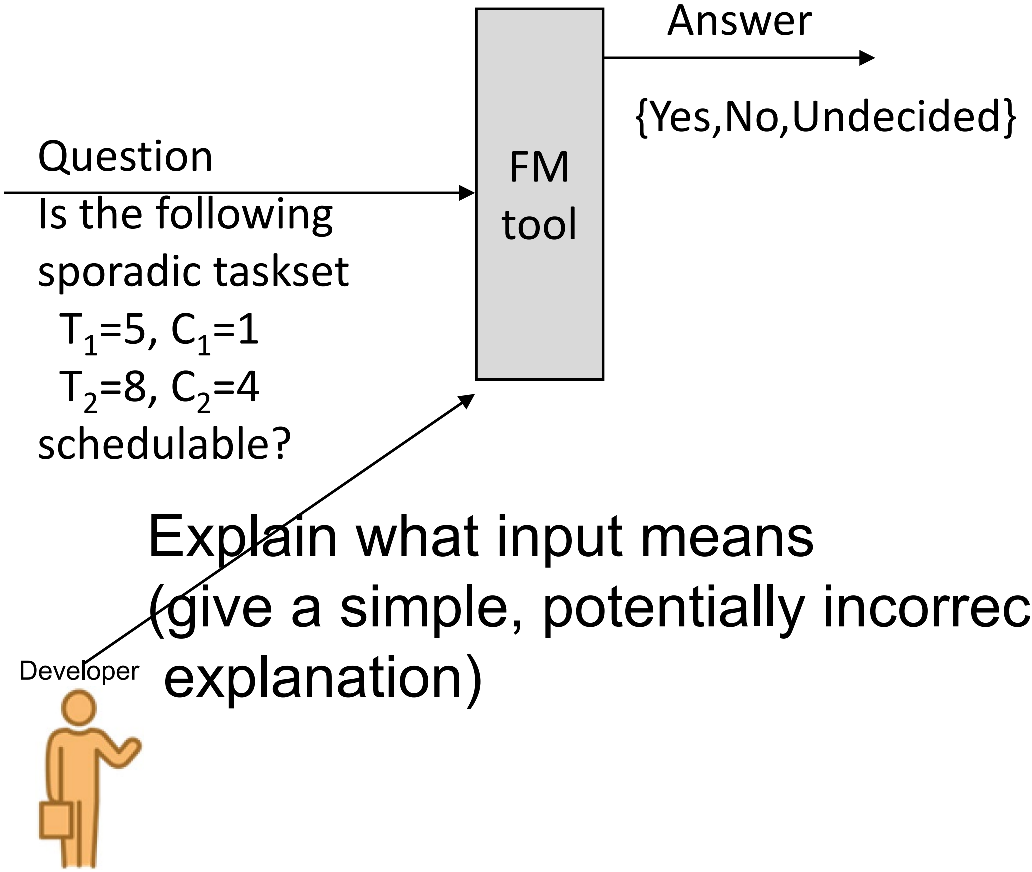
T_1=5, C_1=1

T_2=8, C_2=4

schedulable?

Set default input

Explainability of Input: Real-Time Scheduling



Question

Is the following sporadic taskset

T_1=5, C_1=1

T_2=8, C_2=4

schedulable?

FM tool

Answer

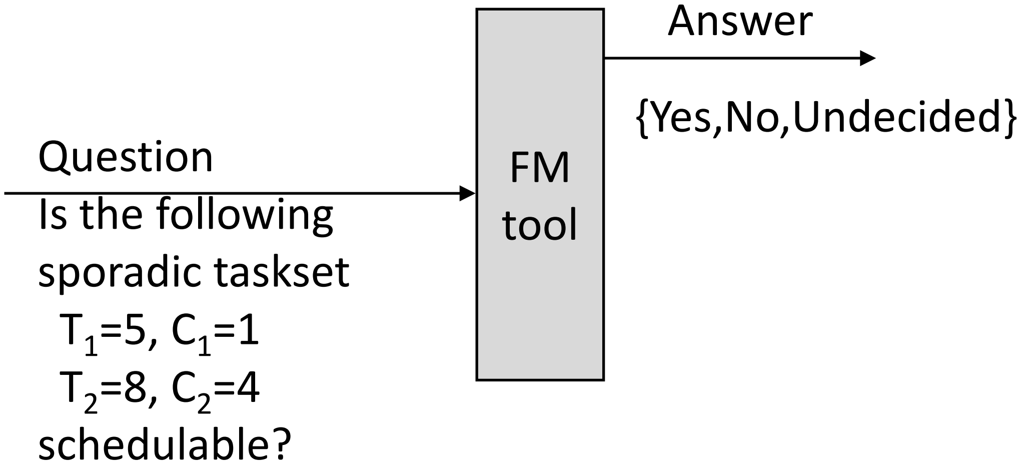
{Yes,No,Undecided}

Explain what input means
(give a simple, potentially incorrect, explanation)

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5
  while (1) {
    delay_until(t)
    do_work(1)
    t=t+5
  }

Thread 2
  t=now()+8
  while (1) {
    delay_until(t)
    do_work(4)
    t=t+8
  }
```



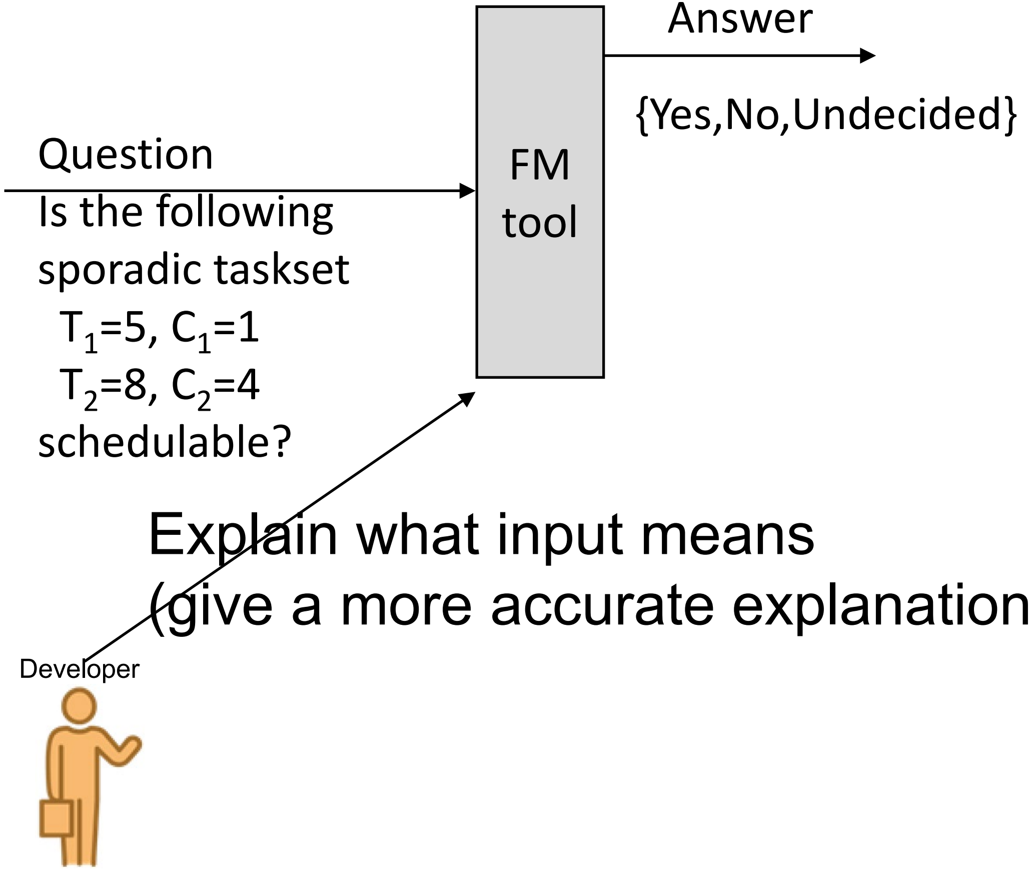
Is the following sporadic taskset

$T_1=5,\ C_1=1$

$T_2=8,\ C_2=4$

schedulable?

Explainability of Input: Real-Time Scheduling



Answer

{Yes,No,Undecided}

Question

Is the following sporadic taskset

$T_1=5$, $C_1=1$

$T_2=8$, $C_2=4$

schedulable?

FM tool

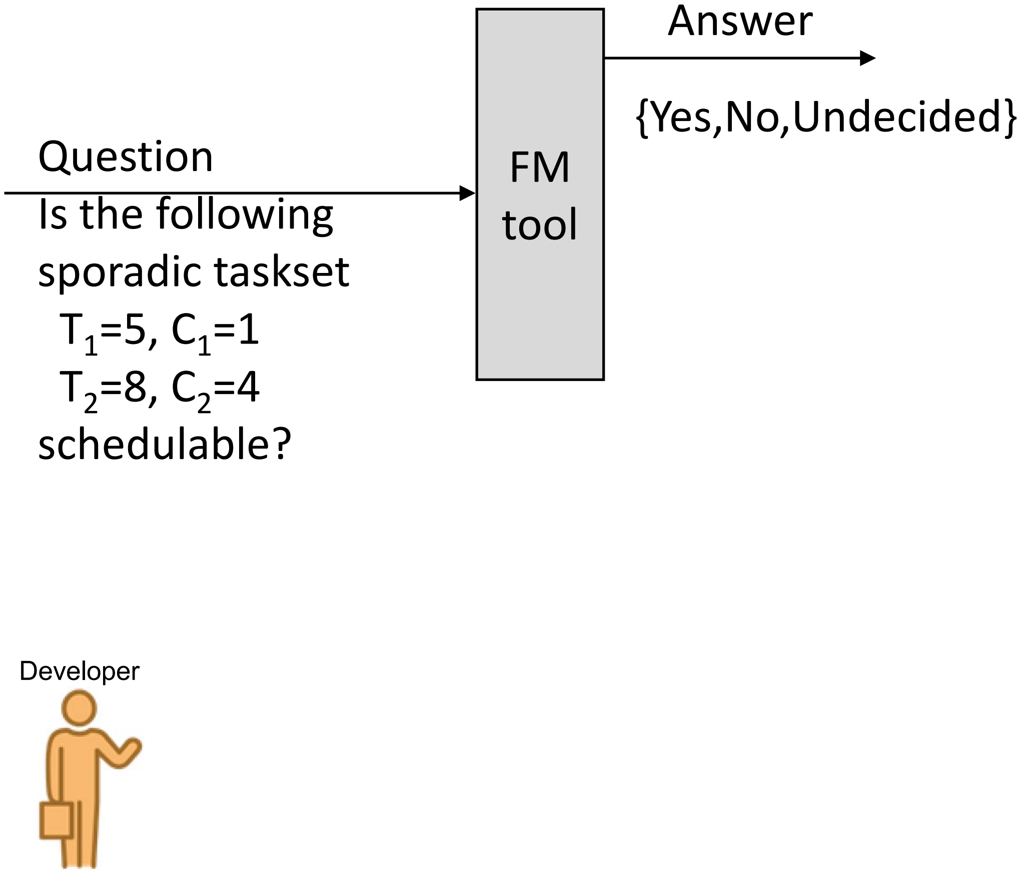
Developer

Explain what input means (give a more accurate explanation)

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5
  }

Thread 2
  t=now()+8
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8
  }
```



Question

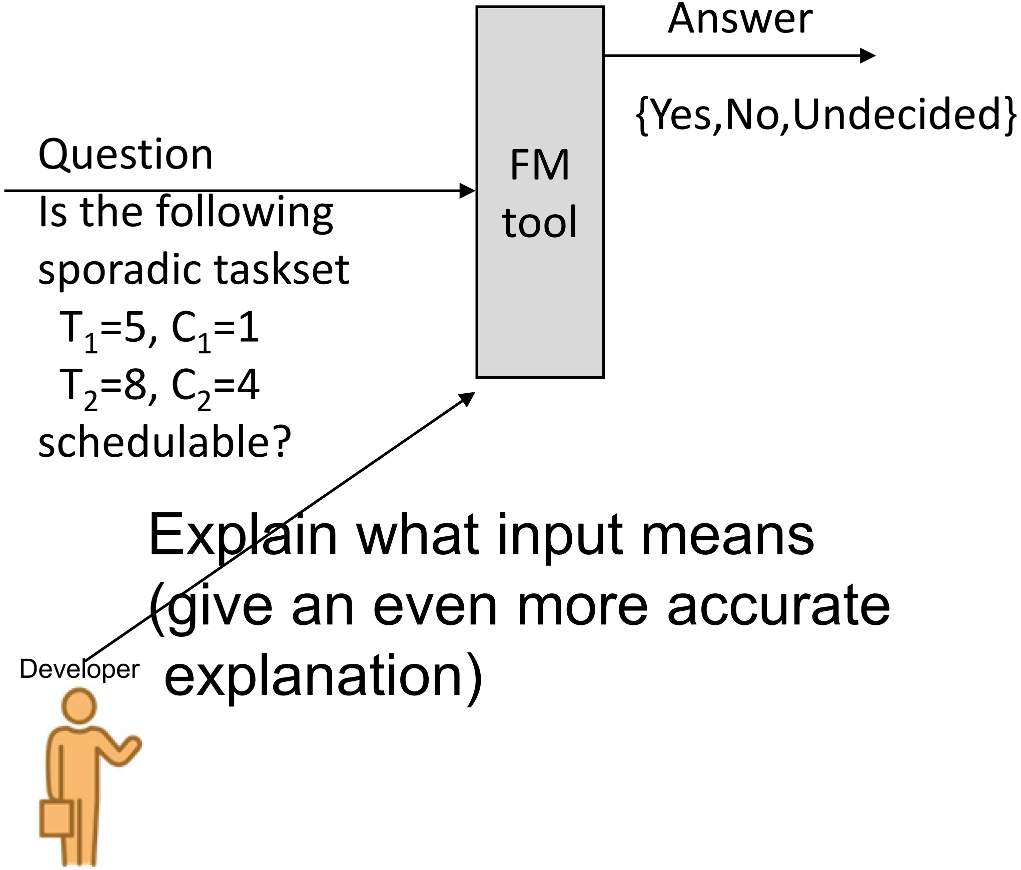
Is the following sporadic taskset

T_1=5, C_1=1

T_2=8, C_2=4

schedulable?

Explainability of Input: Real-Time Scheduling



Question

Is the following sporadic taskset
T_1=5, C_1=1
T_2=8, C_2=4
schedulable?

FM tool

Answer

{Yes,No,Undecided}

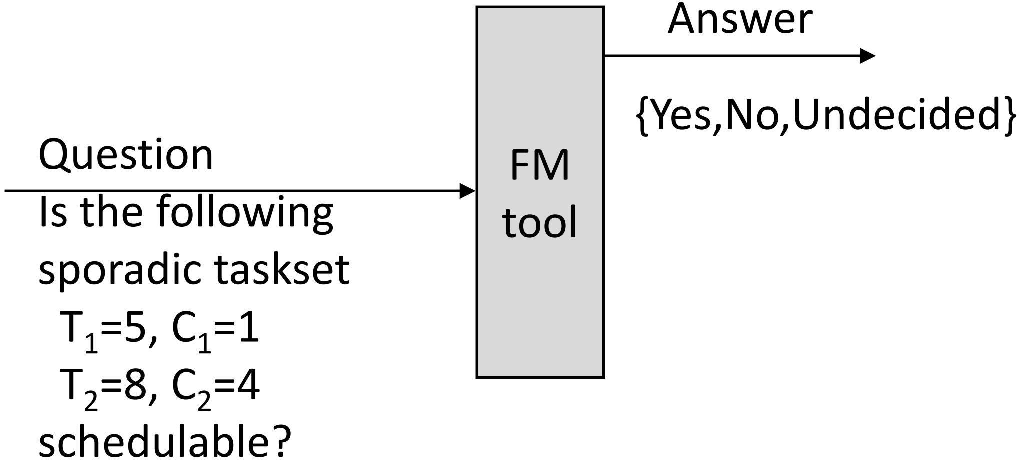
Developer

Explain what input means (give an even more accurate explanation)

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5+rand(0,inf)
  }

Thread 2
  t=now()+8+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8+rand(0,inf)
  }
```



Question

Is the following sporadic taskset

$T_1=5$, $C_1=1$

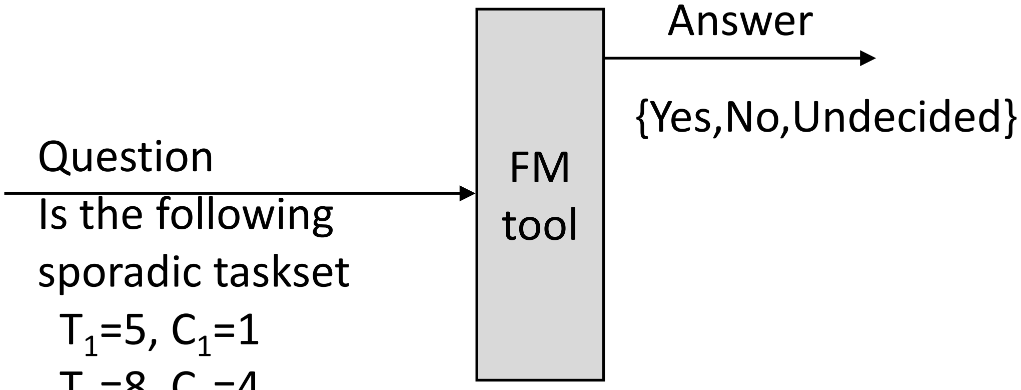
$T_2=8$, $C_2=4$

schedulable?

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5+rand(0,inf)
  }

Thread 2
  t=now()+8+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8+rand(0,inf)
  }
```

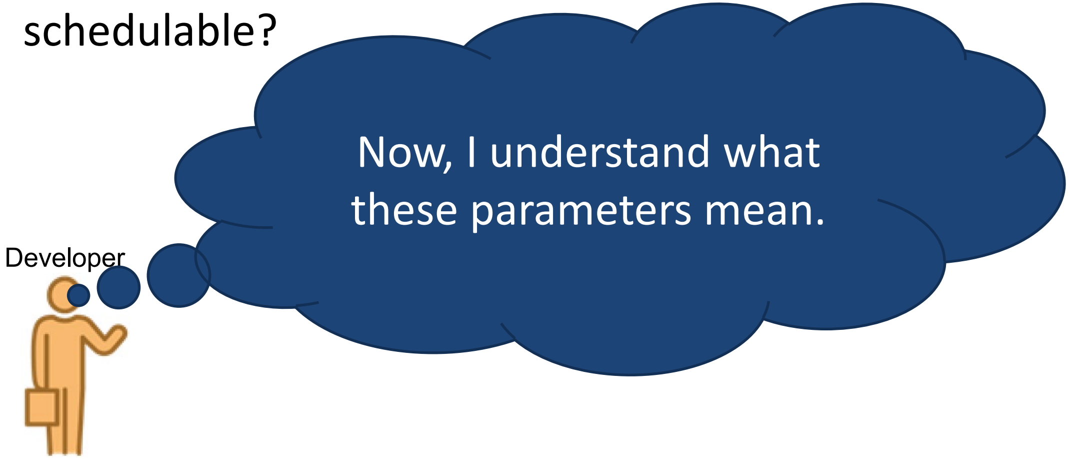


Is the following sporadic taskset

$T_1=5,\ C_1=1$

$T_2=8,\ C_2=4$

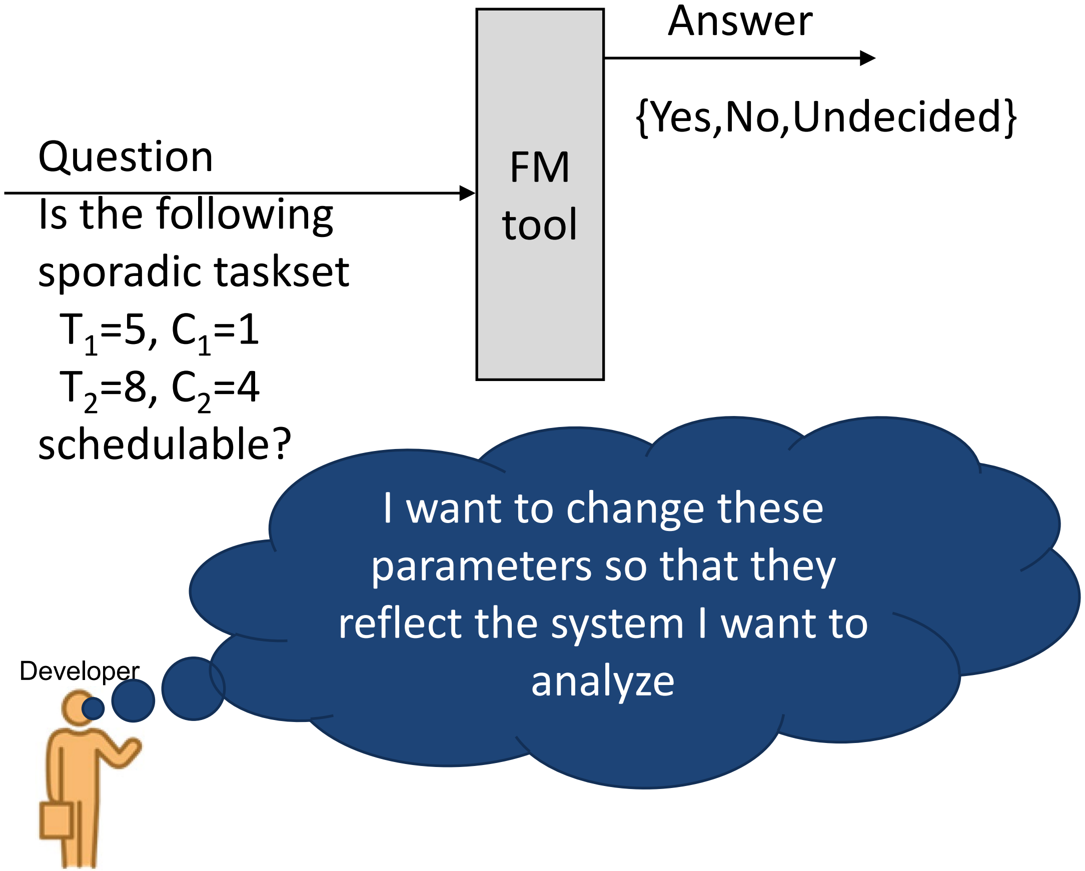
schedulable?



Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5+rand(0,inf)
  }

Thread 2
  t=now()+8+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8+rand(0,inf)
  }
```



Question

Is the following sporadic taskset

$T_1=5$, $C_1=1$

$T_2=8$, $C_2=4$

schedulable?

FM tool

Answer

{Yes,No,Undecided}

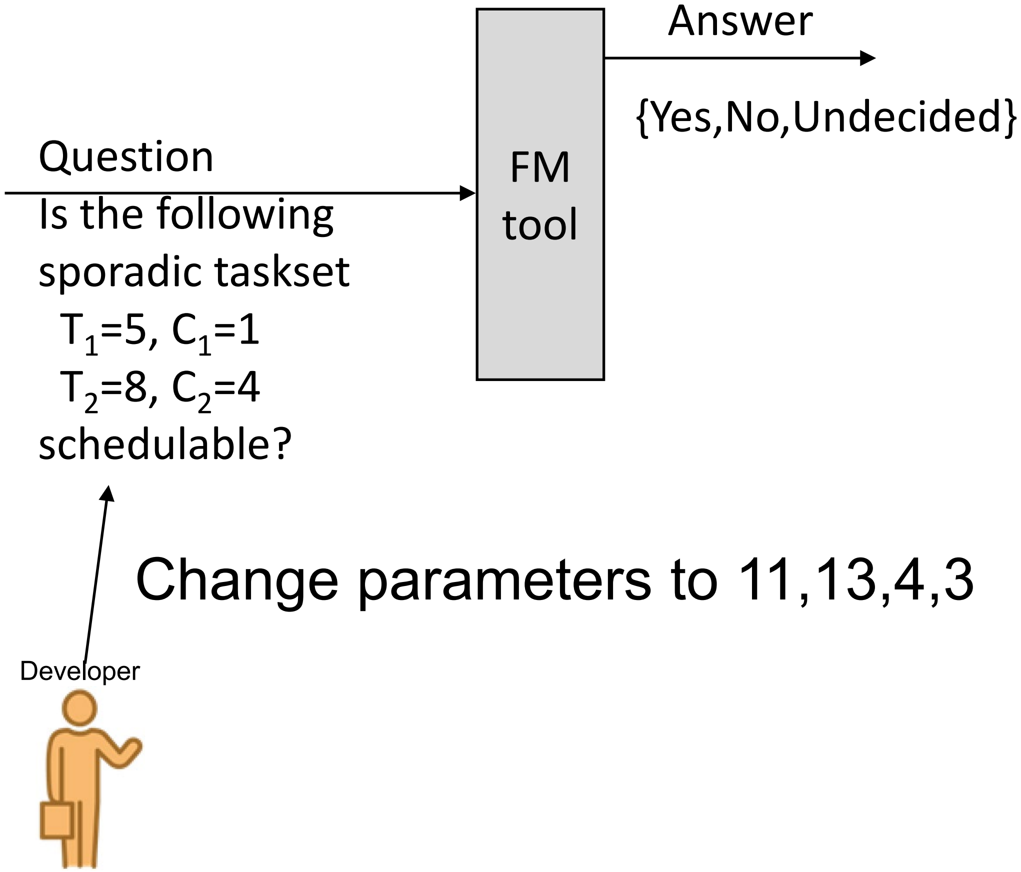
Developer

I want to change these parameters so that they reflect the system I want to analyze

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5+rand(0,inf)
  }

Thread 2
  t=now()+8+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8+rand(0,inf)
  }
```



Is the following sporadic taskset

T_1=5, C_1=1

T_2=8, C_2=4

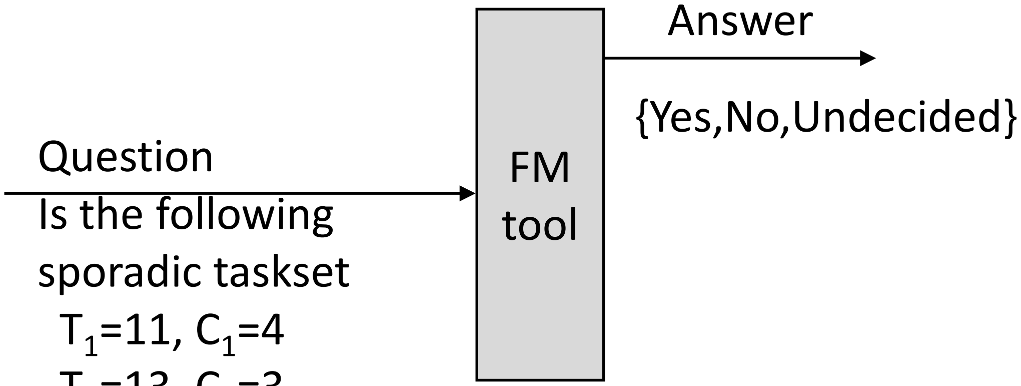
schedulable?

Change parameters to 11,13,4,3

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5+rand(0,inf)
  }

Thread 2
  t=now()+8+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8+rand(0,inf)
  }
```



Is the following sporadic taskset

T_1=11, C_1=4

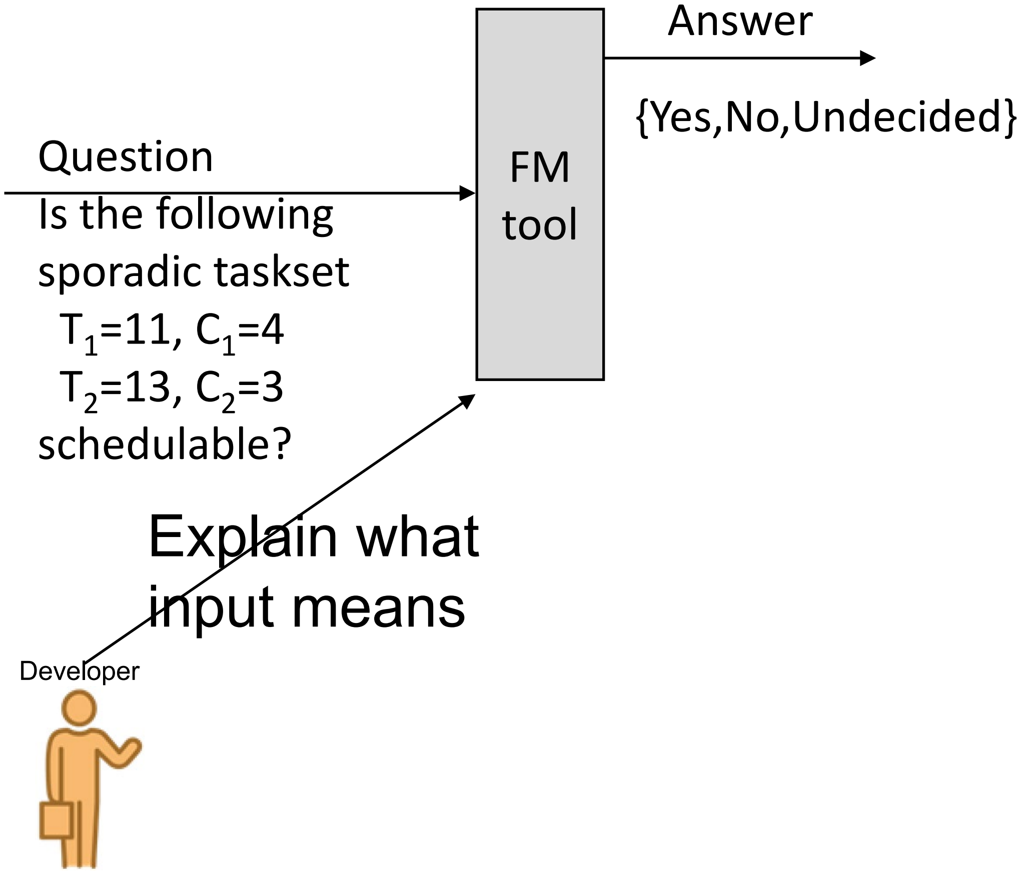
T_2=13, C_2=3

schedulable?

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+5+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,1))
    t=t+5+rand(0,inf)
  }

Thread 2
  t=now()+8+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+8+rand(0,inf)
  }
```



Answer

{Yes,No,Undecided}

Question

Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

schedulable?

FM tool

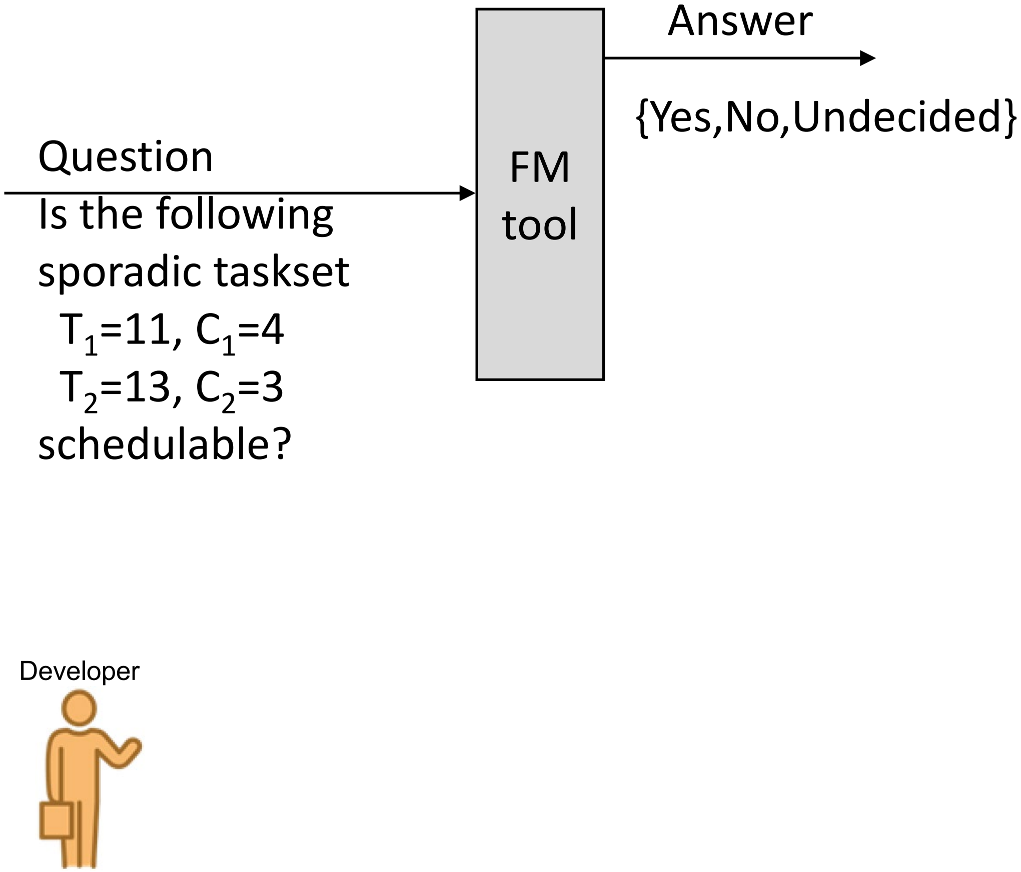
Explain what input means

Developer

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+11+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+11+rand(0,inf)
  }

Thread 2
  t=now()+13+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,3))
    t=t+13+rand(0,inf)
  }
```



Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

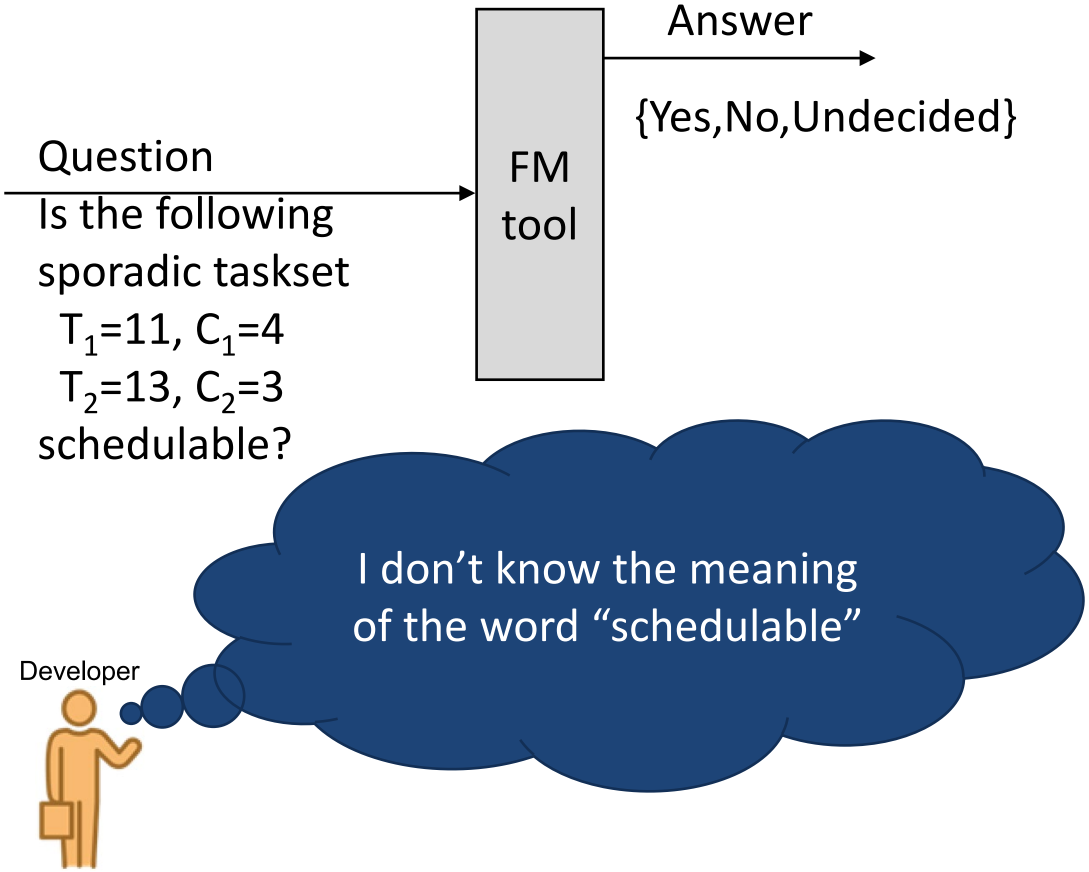
schedulable?

{Yes,No,Undecided}

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+11+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+11+rand(0,inf)
  }

Thread 2
  t=now()+13+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,3))
    t=t+13+rand(0,inf)
  }
```



Question

Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

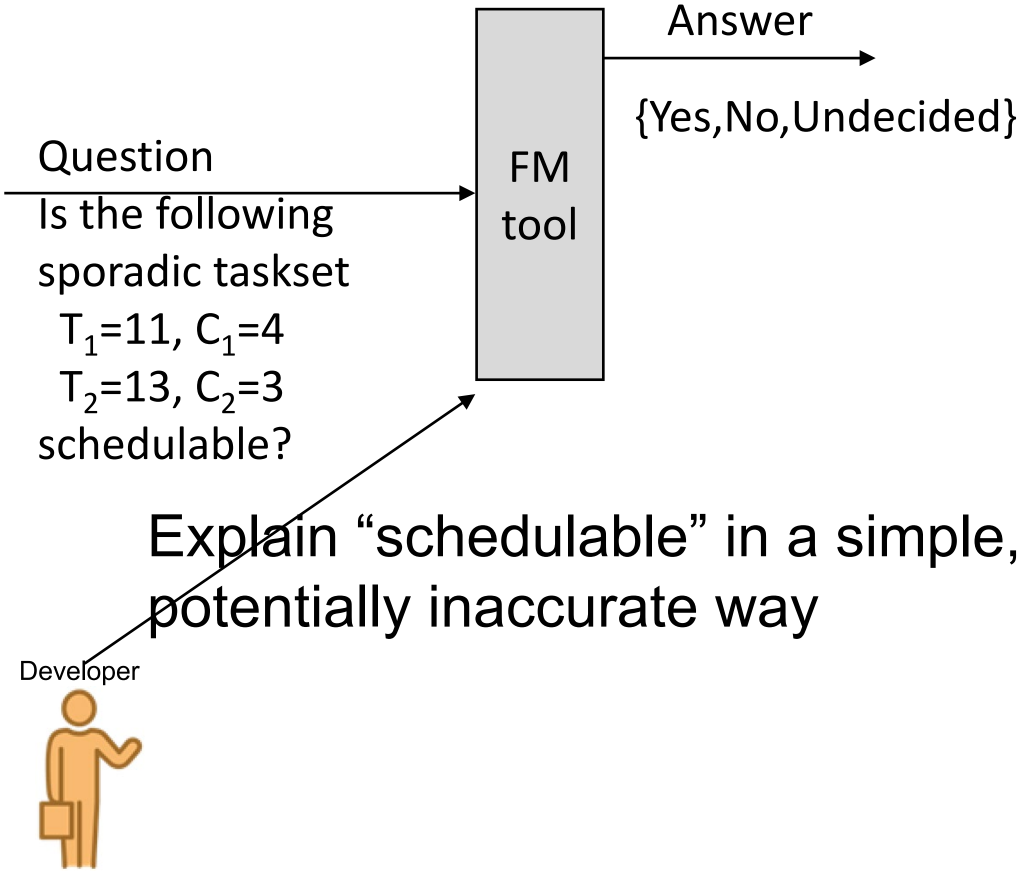
schedulable?

I don’t know the meaning of the word “schedulable”

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+11+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,4))
    t=t+11+rand(0,inf)
  }

Thread 2
  t=now()+13+rand(0,inf)
  while (1) {
    delay_until(t)
    do_work(rand(0,3))
    t=t+13+rand(0,inf)
  }
```



Answer

{Yes,No,Undecided}

FM tool

Question

Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

schedulable?

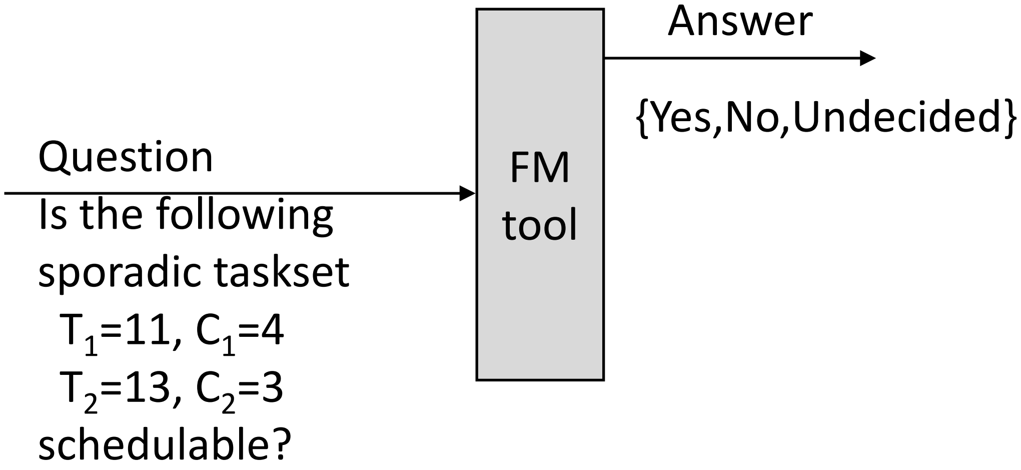
Developer

Explain “schedulable” in a simple, potentially inaccurate way

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+11+rand(0,inf)
  while (1) {
    assert(now()<=t)
    delay_until(t)
    do_work(rand(0,4))
    t=t+11+rand(0,inf)
  }

Thread 2
  t=now()+13+rand(0,inf)
  while (1) {
    assert(now()<=t)
    delay_until(t)
    do_work(rand(0,3))
    t=t+13+rand(0,inf)
  }
```



Question

Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

schedulable?

FM tool

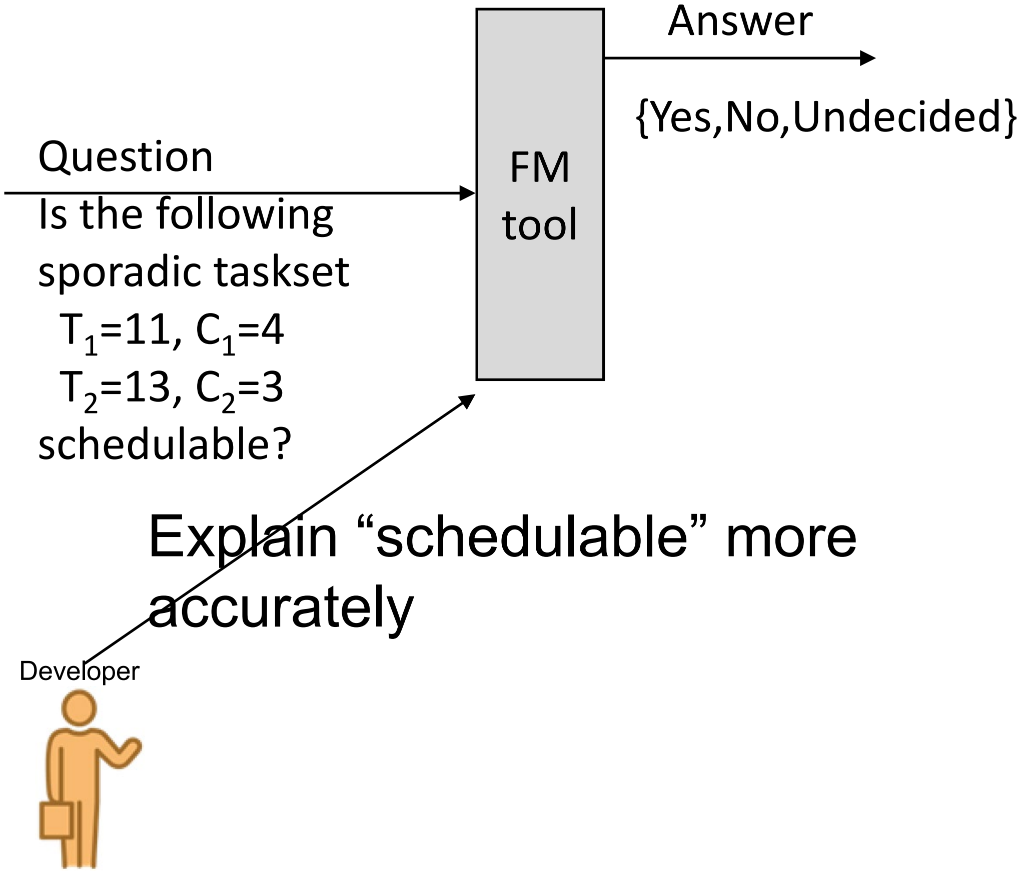
Answer

{Yes,No,Undecided}

Explainability of Input: Real-Time Scheduling

```
Thread 1
  t=now()+11+rand(0,inf)
  while (1) {
    assert(now()<=t)
    delay_until(t)
    do_work(rand(0,4))
    t=t+11+rand(0,inf)
  }

Thread 2
  t=now()+13+rand(0,inf)
  while (1) {
    assert(now()<=t)
    delay_until(t)
    do_work(rand(0,3))
    t=t+13+rand(0,inf)
  }
```



Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

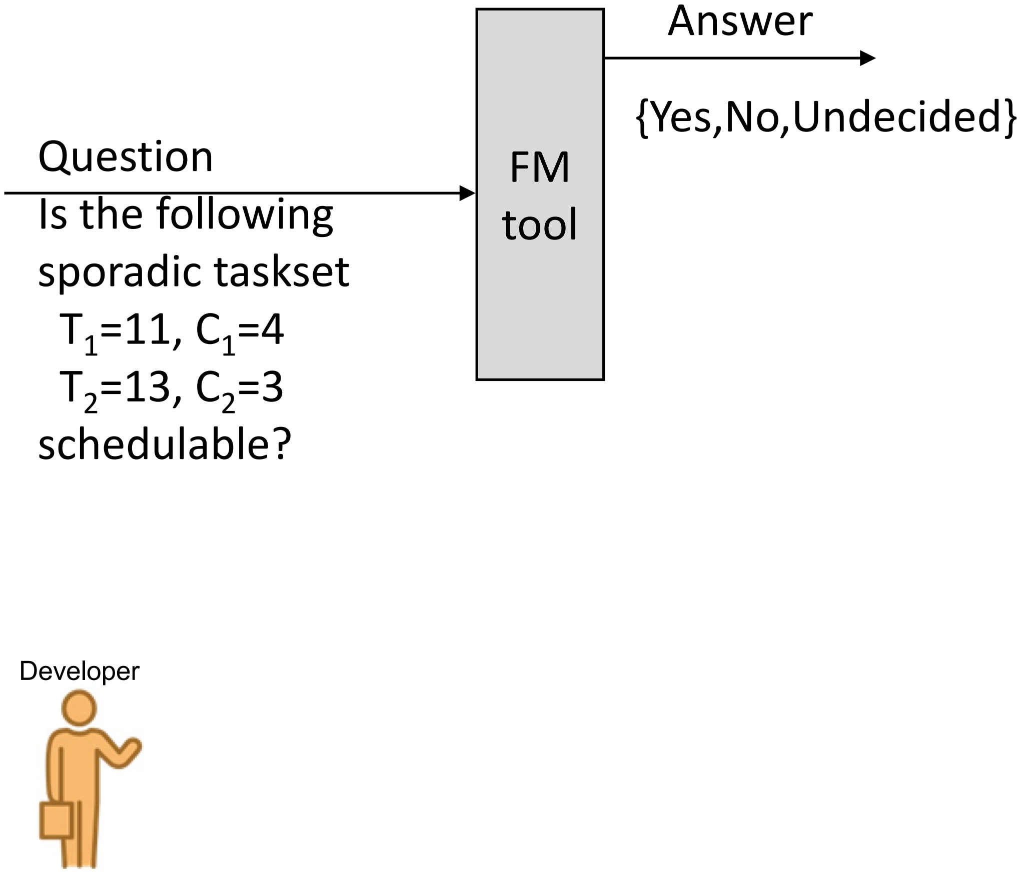
schedulable?

Explain “schedulable” more accurately

Explainability of Input: Real-Time Scheduling

```
Thread 1
  gap=rand(0,inf)
  t=now()+11+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,4))
    gap=rand(0,inf)
    t=t+11+gap
  }

Thread 2
  gap=rand(0,inf)
  t=now()+13+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,3))
    gap=rand(0,inf)
    t=t+13+gap
  }
```



Question

Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

schedulable?

FM tool

Answer

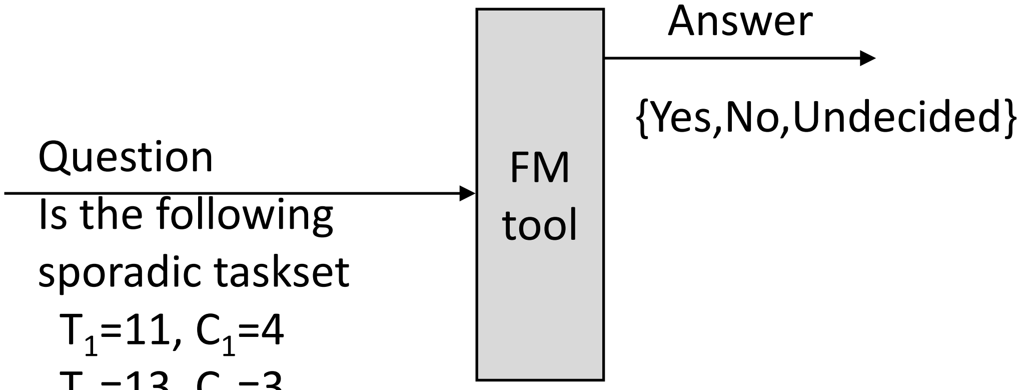
{Yes,No,Undecided}

Developer

Explainability of Input: Real-Time Scheduling

```
Thread 1
  gap=rand(0,inf)
  t=now()+11+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,4))
    gap=rand(0,inf)
    t=t+11+gap
  }

Thread 2
  gap=rand(0,inf)
  t=now()+13+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,3))
    gap=rand(0,inf)
    t=t+13+gap
  }
```

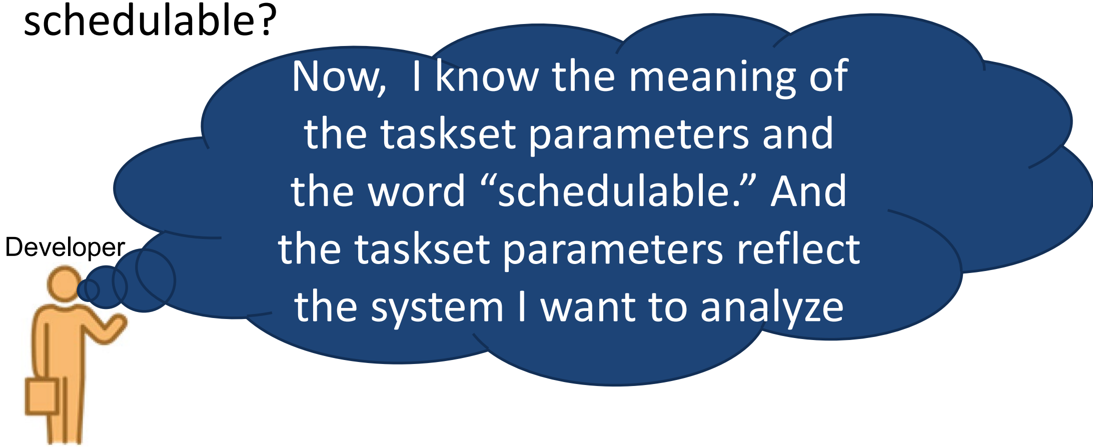


Is the following sporadic taskset

T_1=11, C_1=4

T_2=13, C_2=3

schedulable?

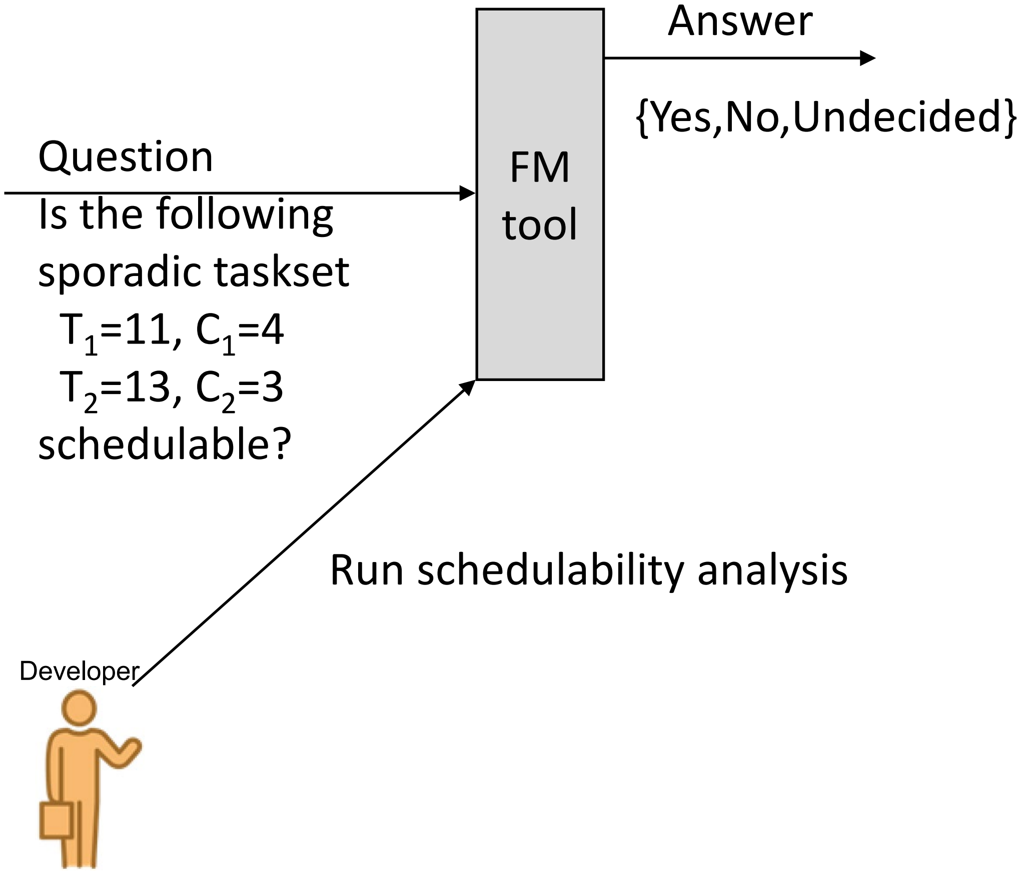


Now, I know the meaning of the taskset parameters and the word “schedulable.” And the taskset parameters reflect the system I want to analyze

Explainability of Input: Real-Time Scheduling

```
Thread 1
  gap=rand(0,inf)
  t=now()+11+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,4))
    gap=rand(0,inf)
    t=t+11+gap
  }

Thread 2
  gap=rand(0,inf)
  t=now()+13+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,3))
    gap=rand(0,inf)
    t=t+13+gap
  }
```



Is the following sporadic taskset

T_1=11, C_1=4

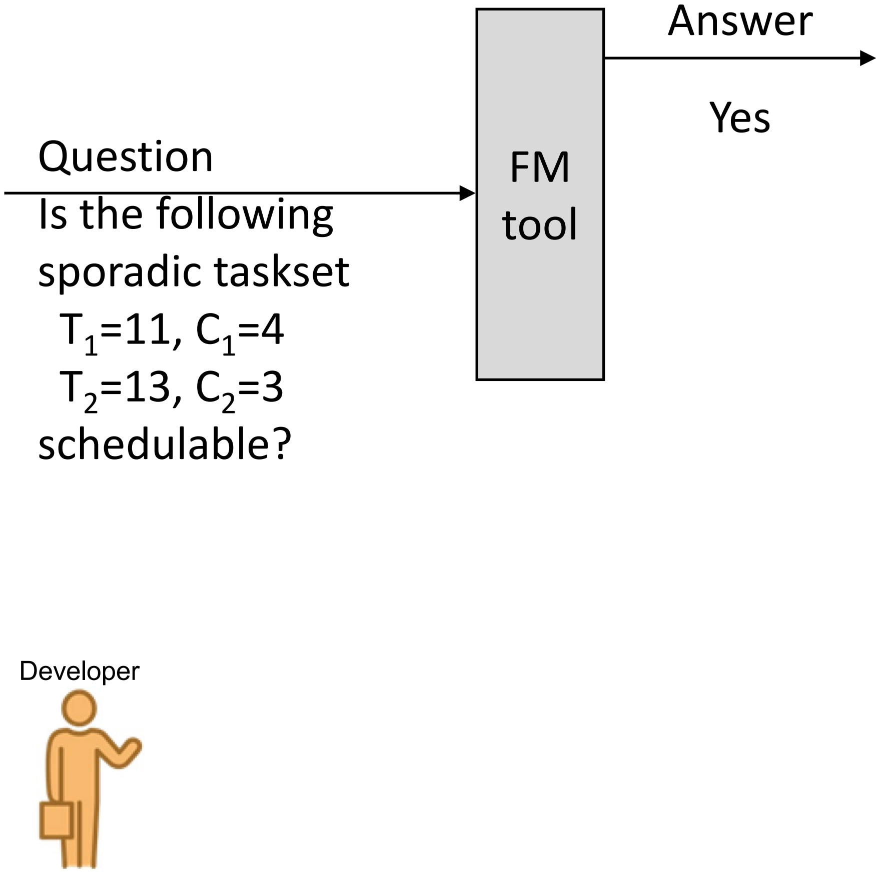
T_2=13, C_2=3

schedulable?

Explainability of Input: Real-Time Scheduling

```
Thread 1
  gap=rand(0,inf)
  t=now()+11+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,4))
    gap=rand(0,inf)
    t=t+11+gap
  }

Thread 2
  gap=rand(0,inf)
  t=now()+13+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,3))
    gap=rand(0,inf)
    t=t+13+gap
  }
```



Yes

Is the following sporadic taskset

T_1=11, C_1=4

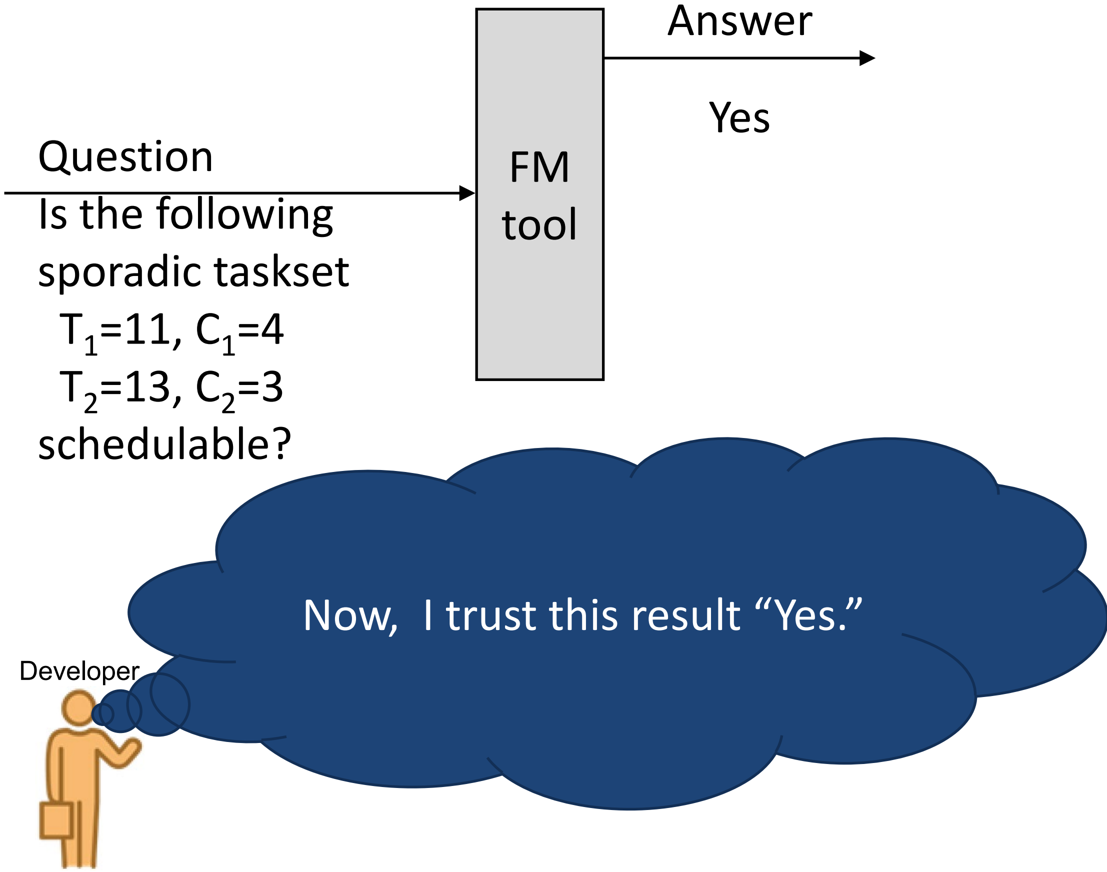
T_2=13, C_2=3

schedulable?

Explainability of Input: Real-Time Scheduling

```
Thread 1
  gap=rand(0,inf)
  t=now()+11+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,4))
    gap=rand(0,inf)
    t=t+11+gap
  }

Thread 2
  gap=rand(0,inf)
  t=now()+13+gap
  while (1) {
    assert(now()<=t-gap)
    delay_until(t)
    do_work(rand(0,3))
    gap=rand(0,inf)
    t=t+13+gap
  }
```



Is the following sporadic taskset

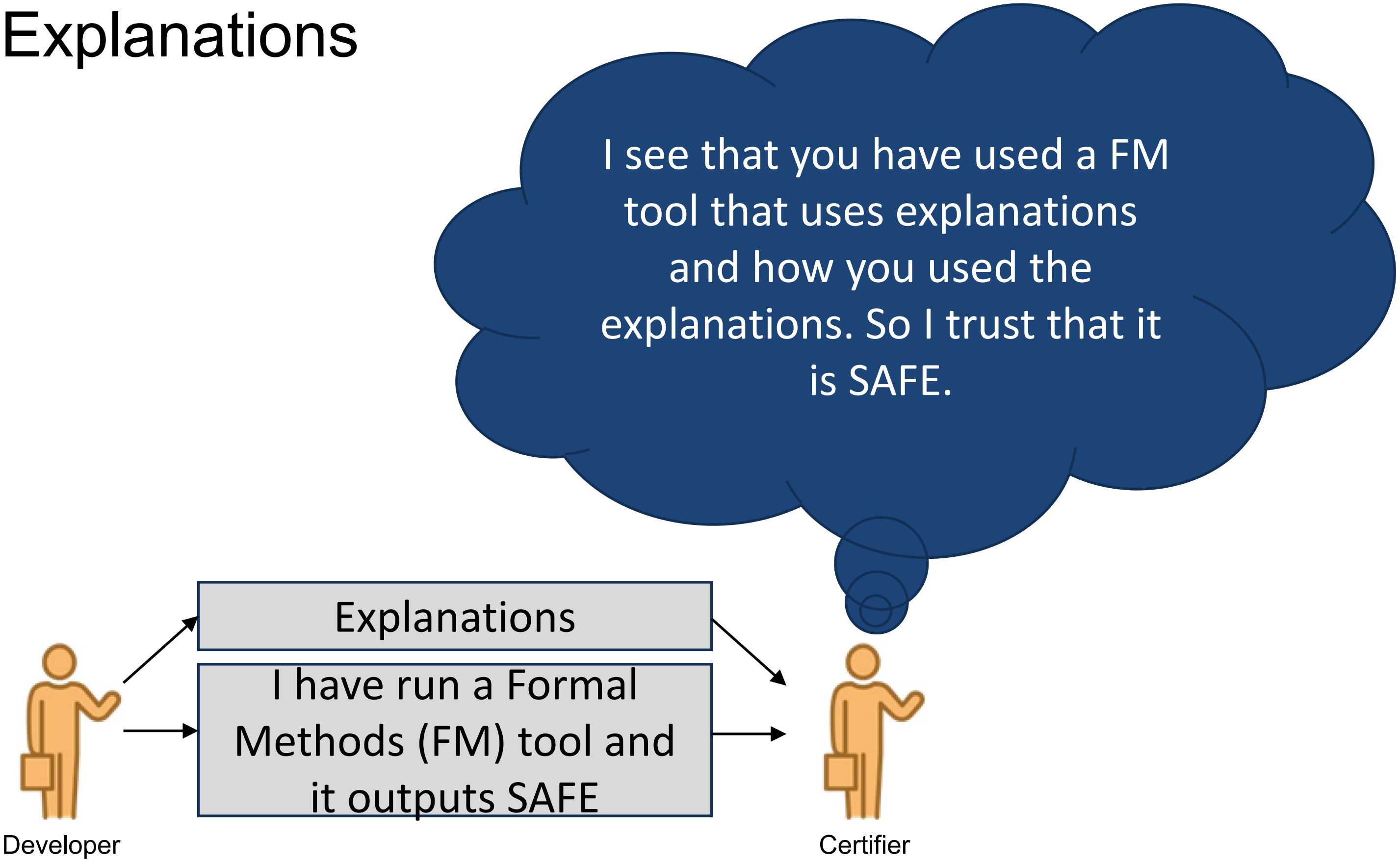
T_1=11, C_1=4

T_2=13, C_2=3

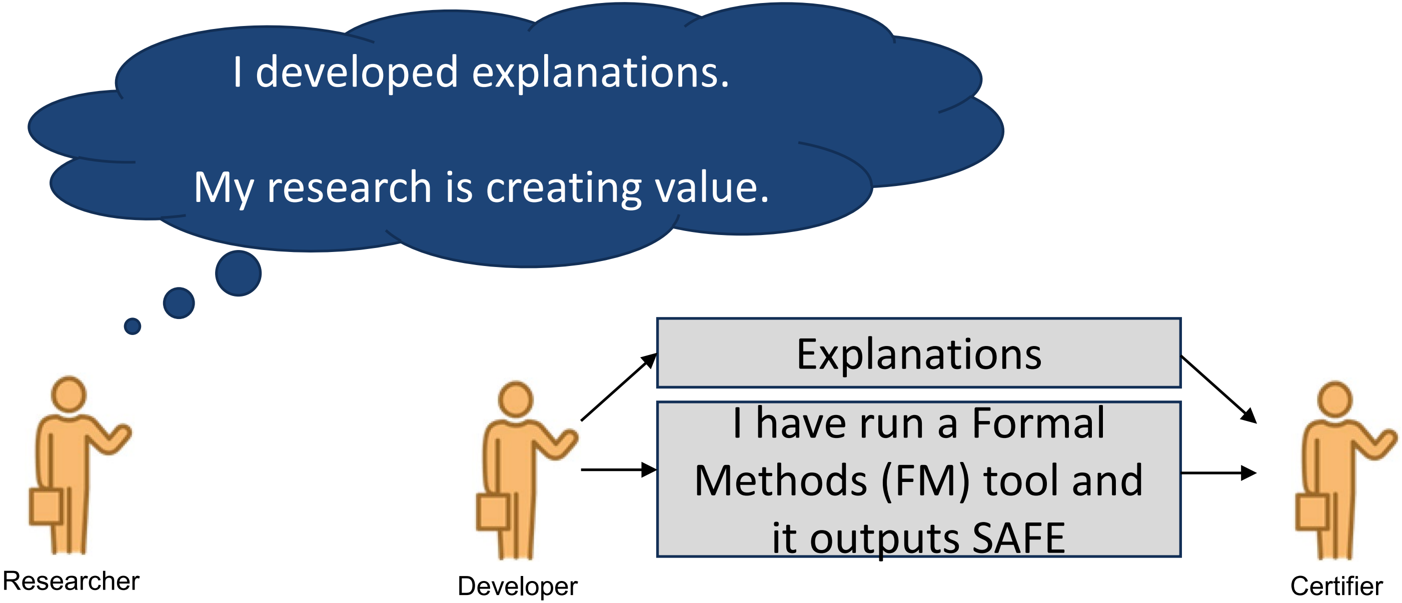
schedulable?

Now, I trust this result “Yes.”

Problem: How to get developers and certifiers to trust FMs
Solution: Use Explanations


I see that you have used a FM tool that uses explanations and how you used the explanations. So I trust that it is SAFE.
Explanations
I have run a Formal Methods (FM) tool and it outputs SAFE
Developer
Certifier

Problem: How to get developers and certifiers to trust FMs
Solution: Use Explanations



I developed explanations.

My research is creating value.

Explanations

I have run a Formal Methods (FM) tool and it outputs SAFE

Conclusion

Explainability of real-time systems and their analysis:

has the potential to make research results valuable to practitioners

is a new, rich, area of research