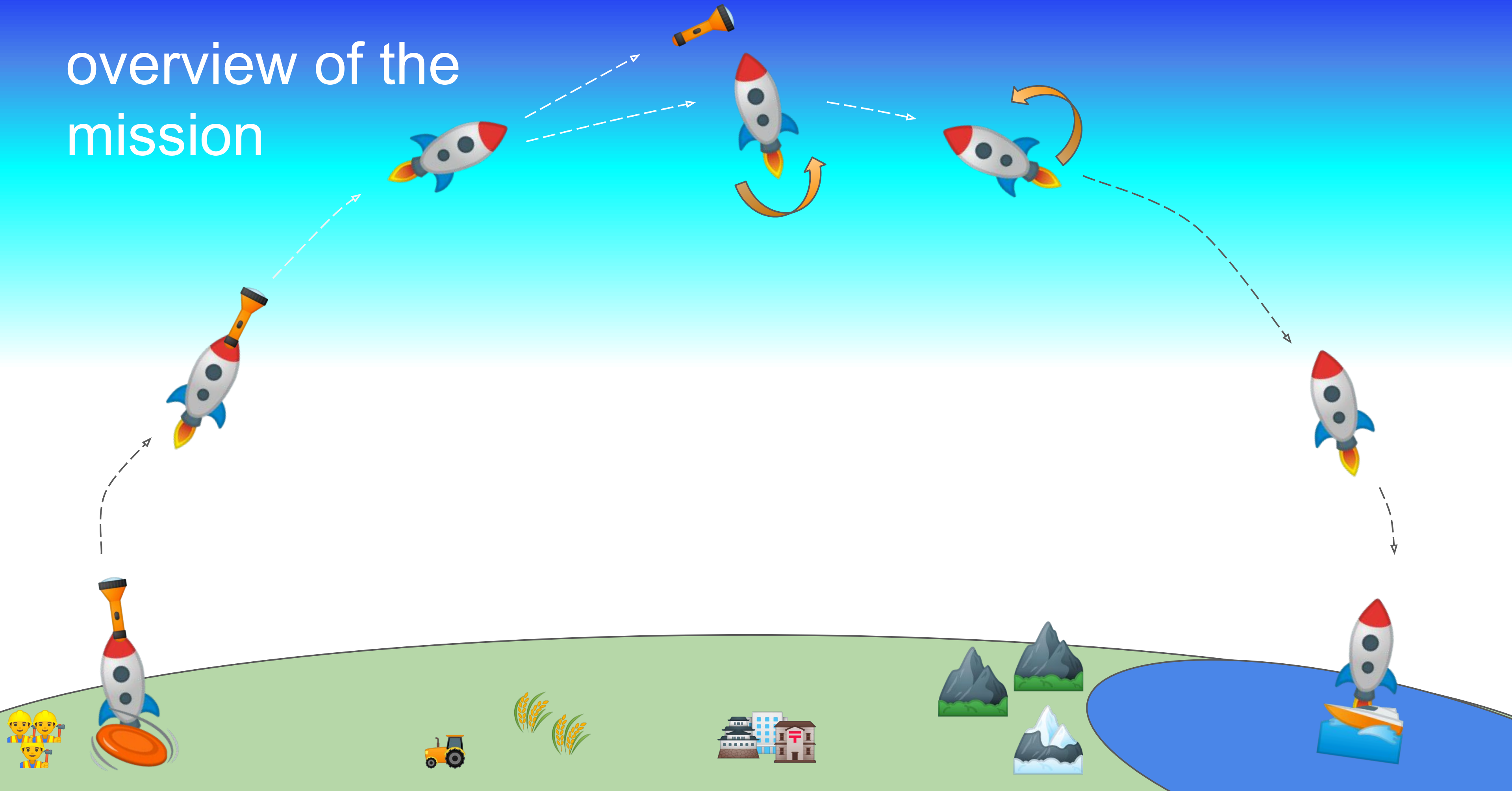
Space Informatics

Week 11: Safety and Reliability of Space System

Computer Science and Communications, University of Luxembourg

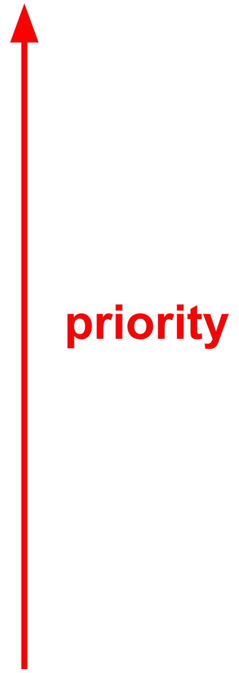
26 November 2019

overview of the mission

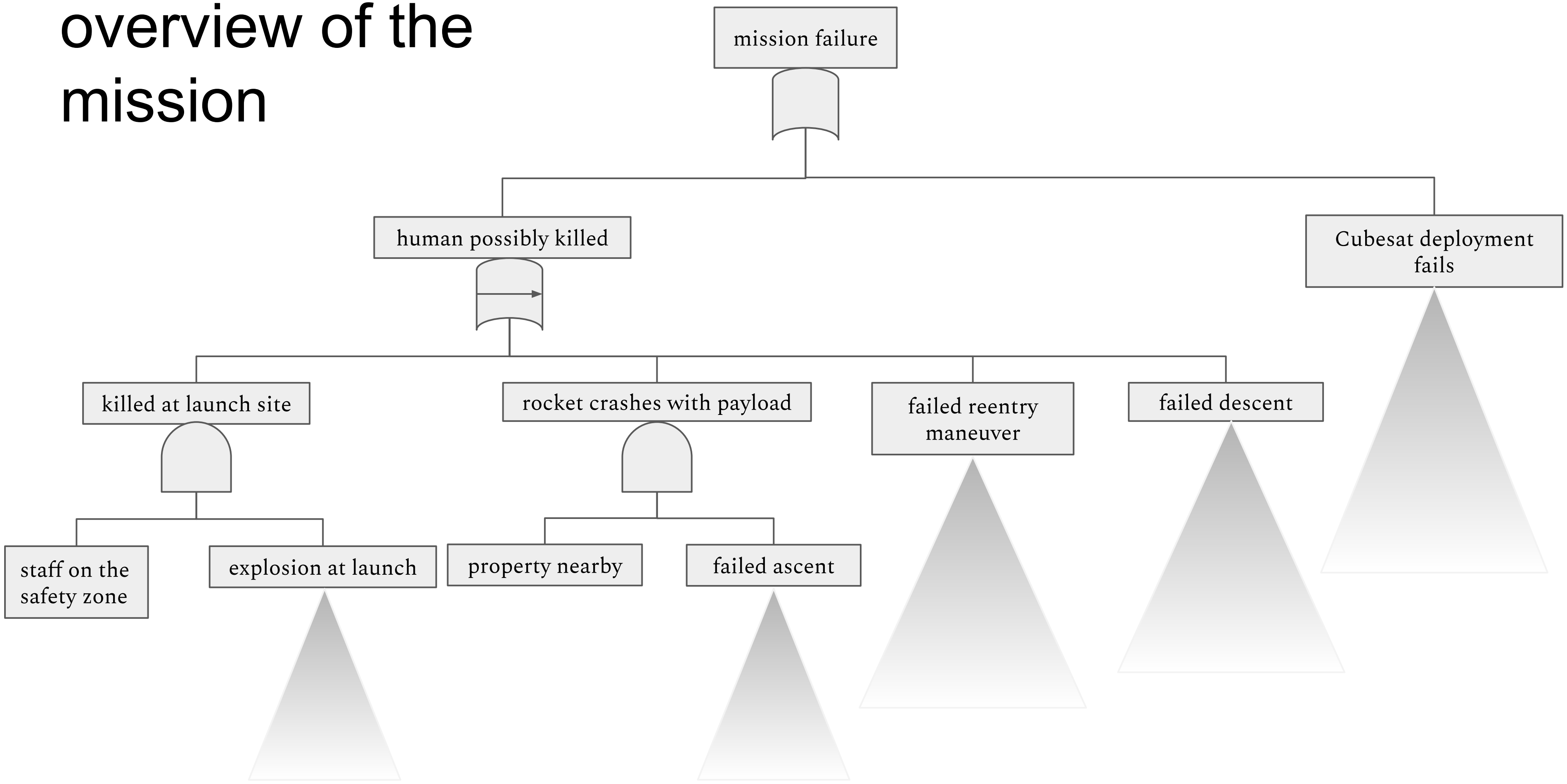


General objectives

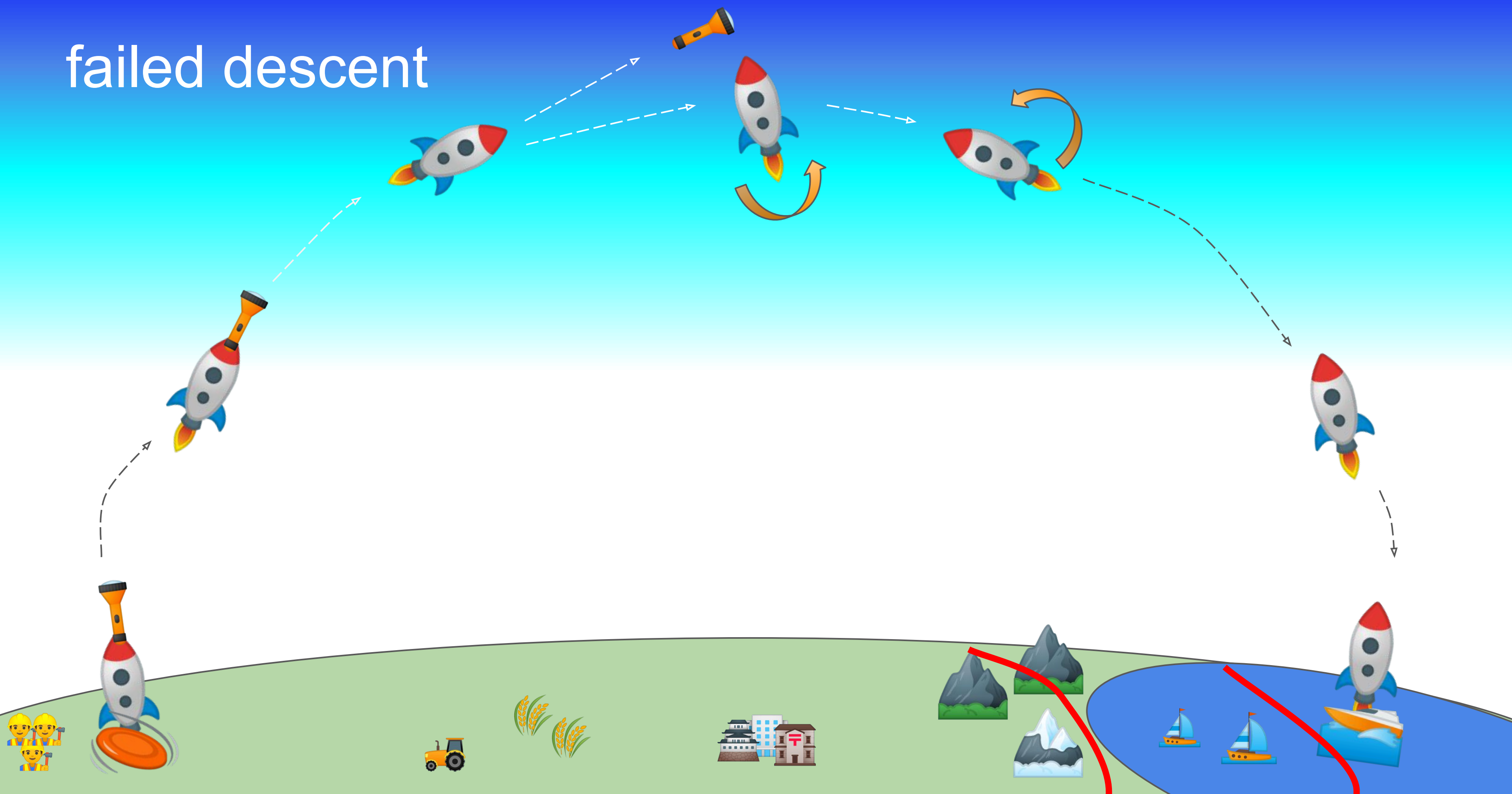
- Critical: no human damages 💀
- High: no property destroyed on the ground 💰
- Medium: fail to put the payload in Low Earth Orbit (LEO)



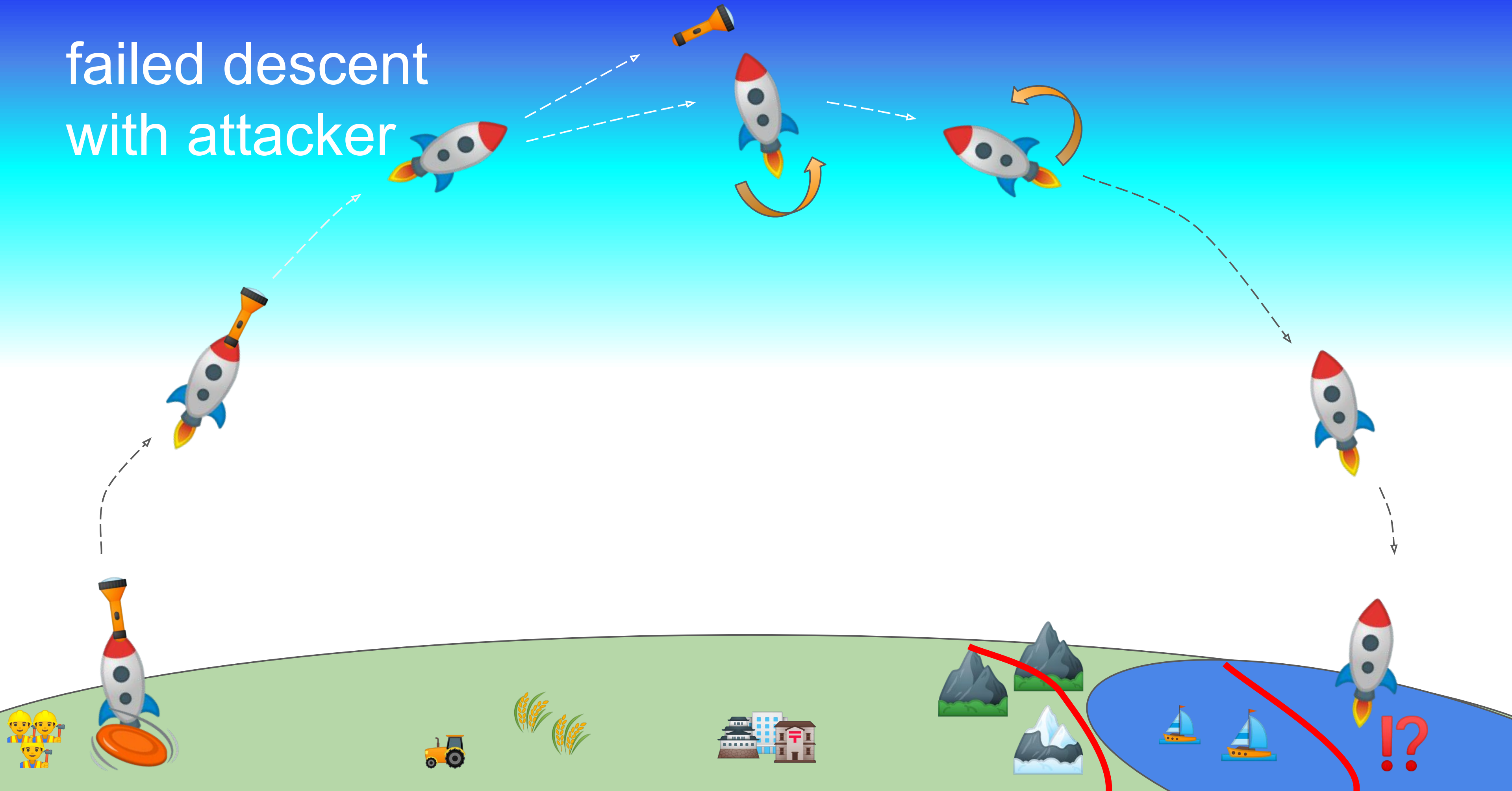
overview of the mission



failed descent

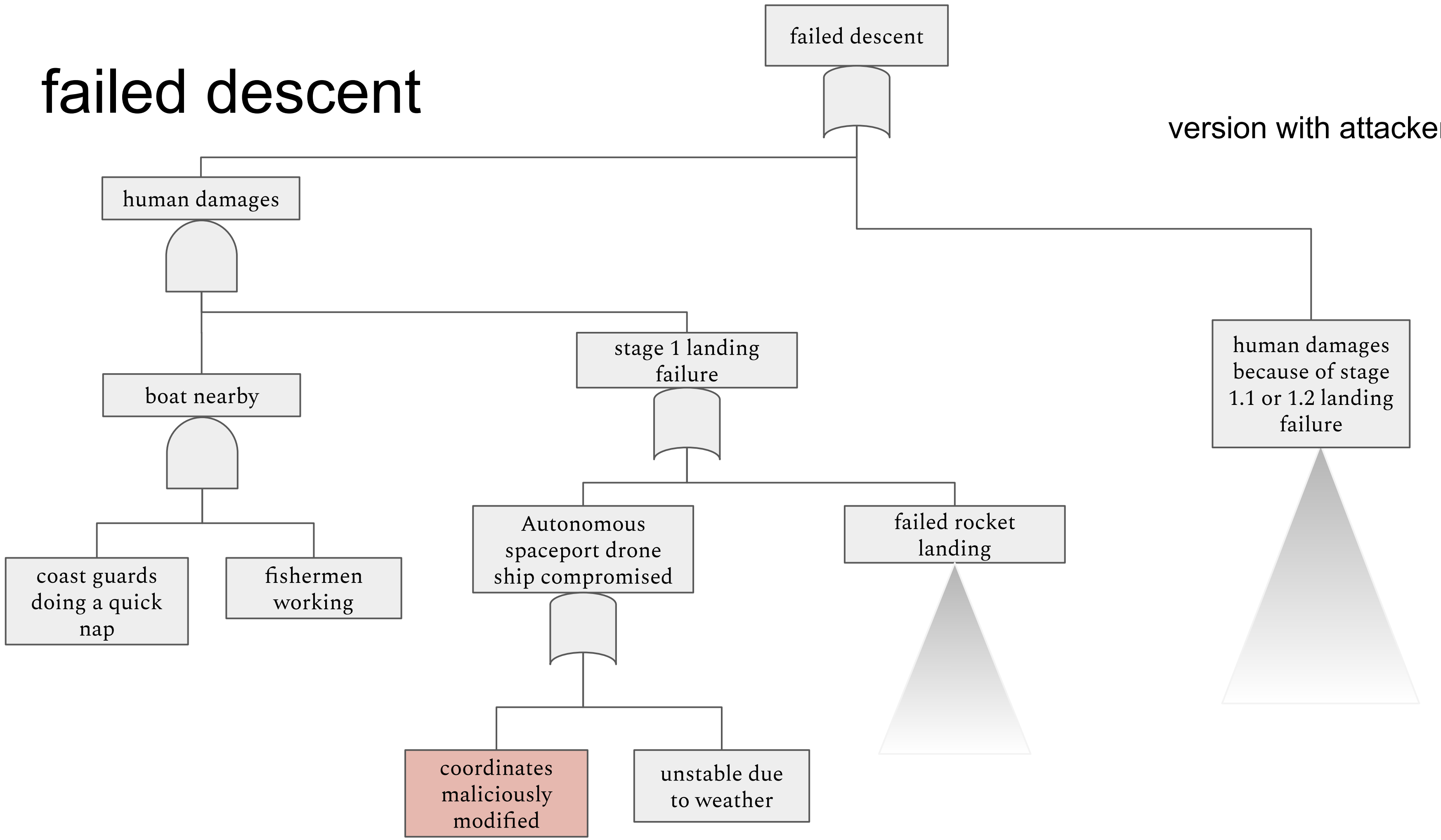


failed descent with attacker

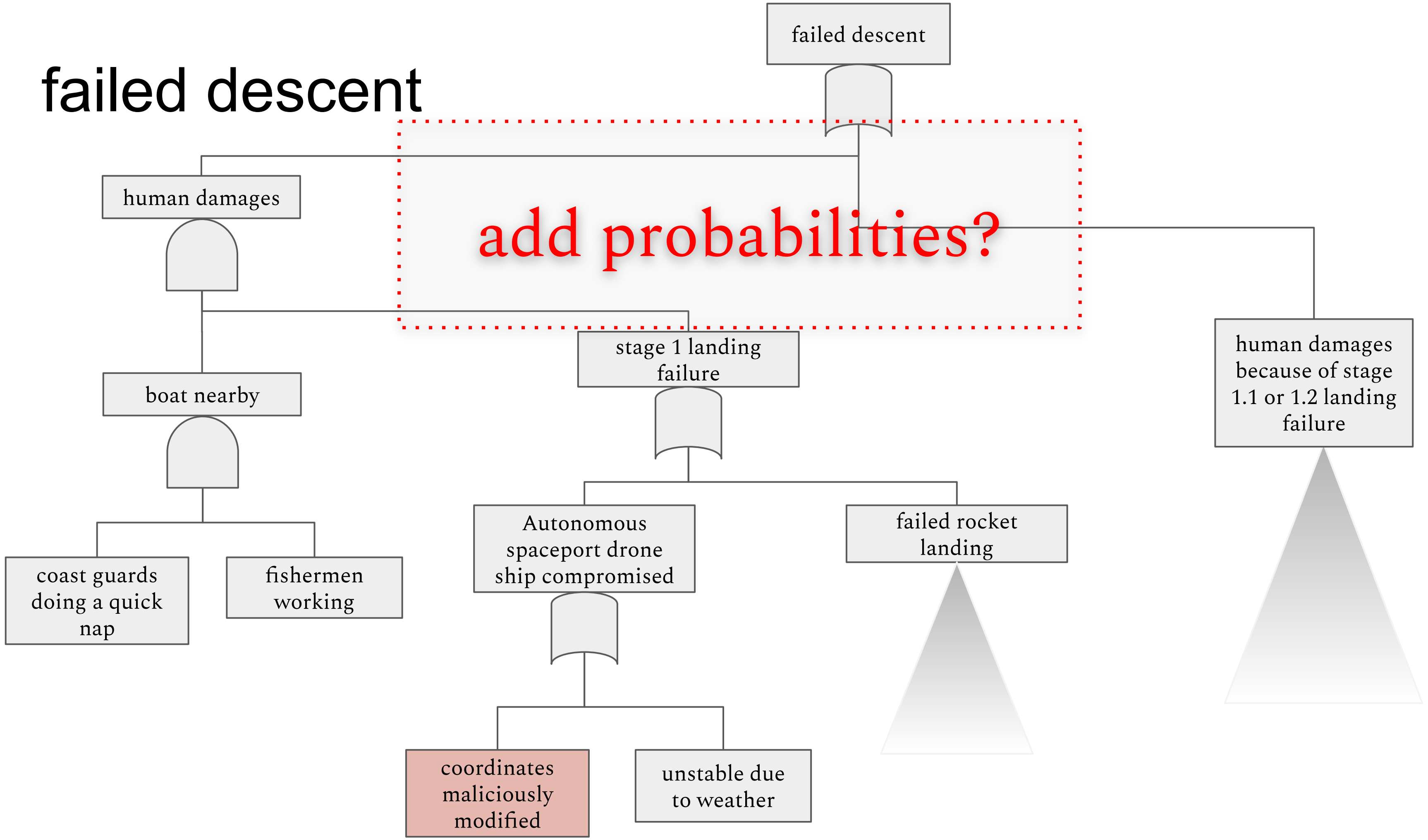


failed descent

version with attacker

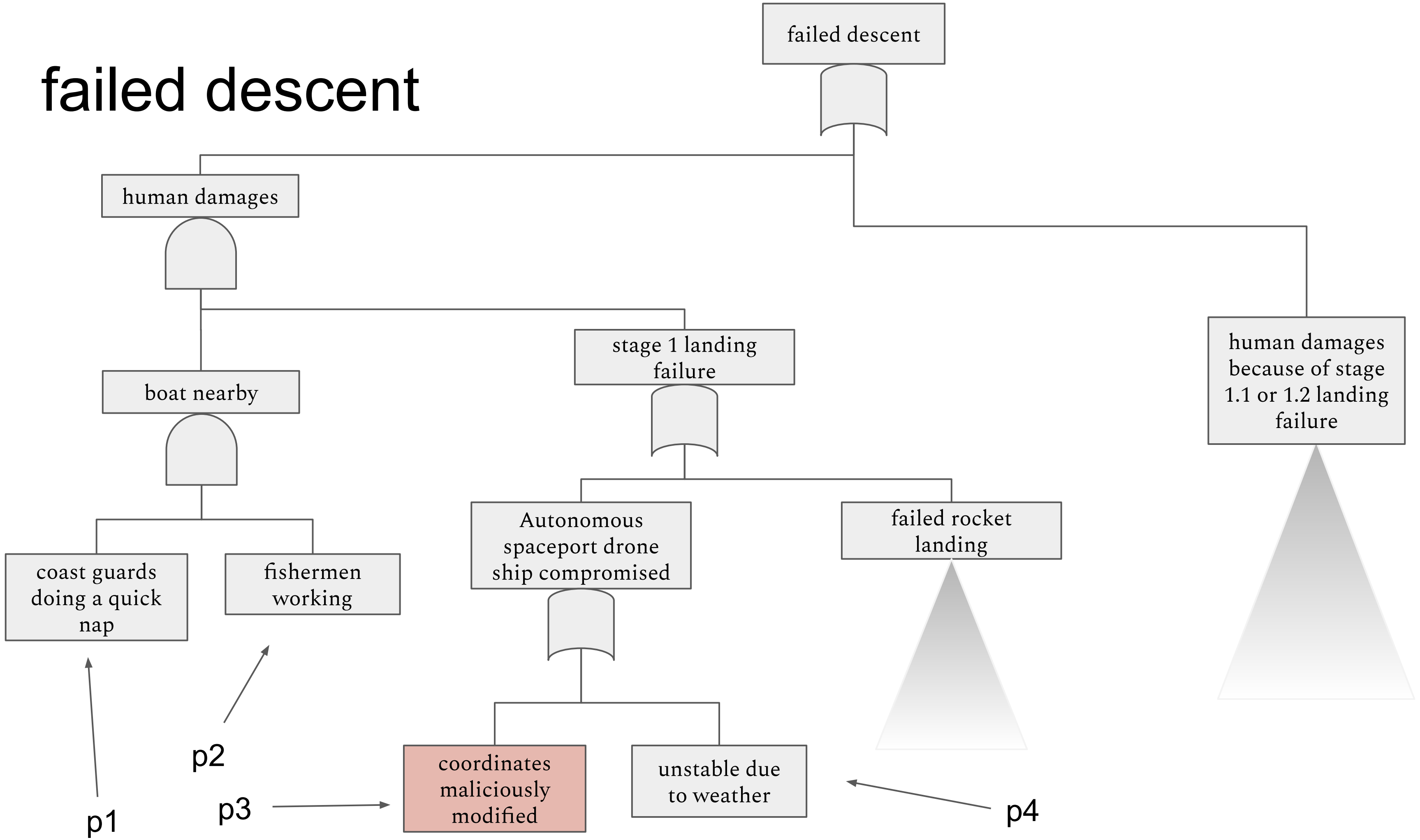

failed descent
human damages
boat nearby
coast guards doing a quick nap
fishermen working
stage 1 landing failure
Autonomous spaceport drone ship compromised
coordinates maliciously modified
unstable due to weather
failed rocket landing
human damages because of stage 1.1 or 1.2 landing failure

failed descent

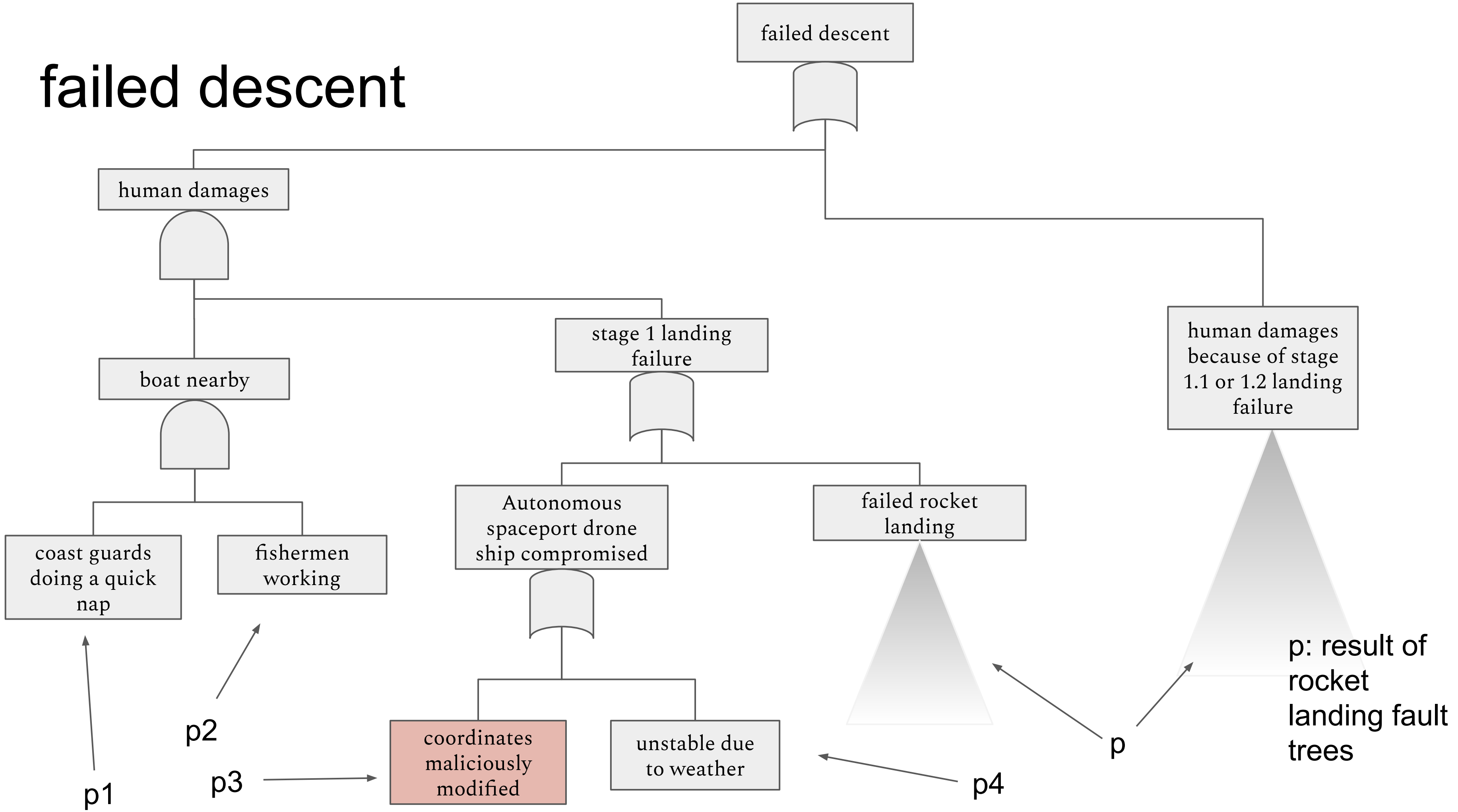

failed descent
human damages
add probabilities?
stage 1 landing failure
human damages because of stage 1.1 or 1.2 landing failure
boat nearby
Autonomous spaceport drone ship compromised
failed rocket landing
coast guards doing a quick nap
fishermen working
coordinates maliciously modified
unstable due to weather

1. Fault tree analysis: probabilistic events
2. Fault tree analysis: costs and damages computation?
3. group work

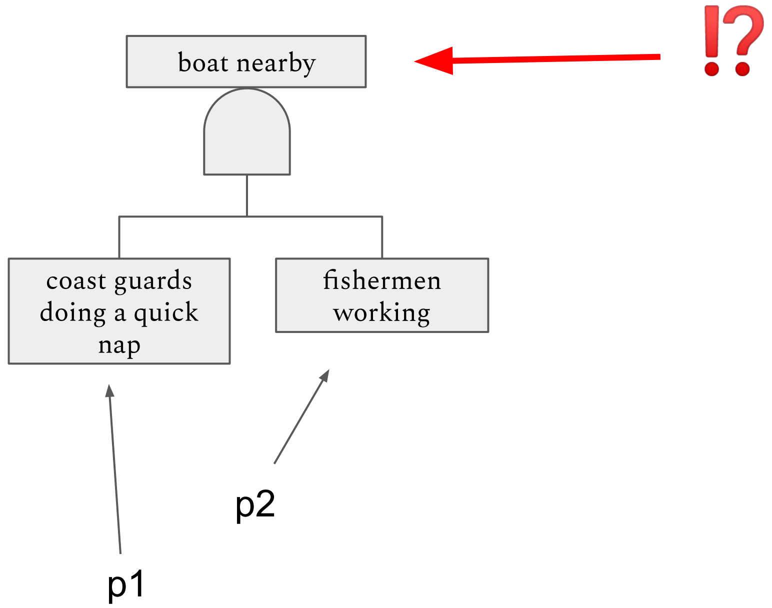
failed descent


failed descent
human damages
boat nearby
stage 1 landing failure
human damages because of stage 1.1 or 1.2 landing failure
coast guards doing a quick nap
fishermen working
Autonomous spaceport drone ship compromised
failed rocket landing
coordinates maliciously modified
unstable due to weather
p1
p2
p3
p4

failed descent

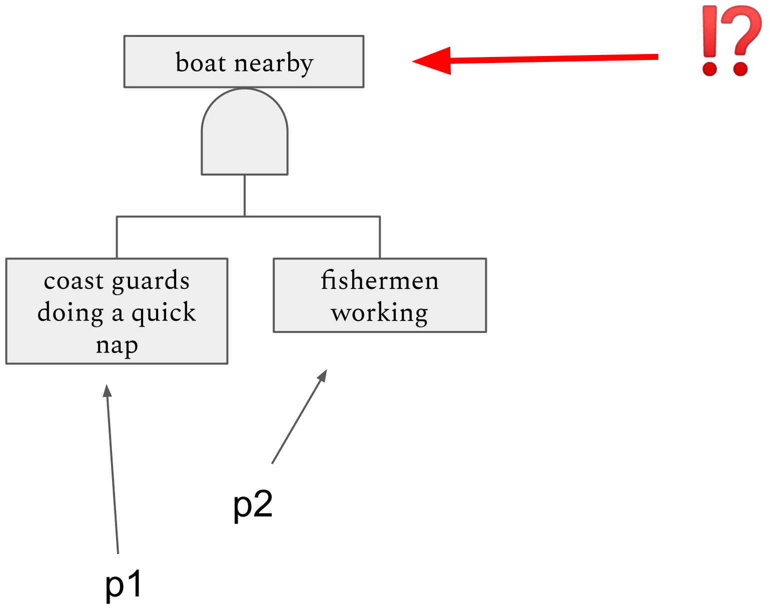

failed descent
human damages
boat nearby
stage 1 landing failure
human damages because of stage 1.1 or 1.2 landing failure
coast guards doing a quick nap
fishermen working
Autonomous spaceport drone ship compromised
failed rocket landing
coordinates maliciously modified
unstable due to weather
p1
p2
p3
p4
p
p: result of rocket landing fault trees

AND gate



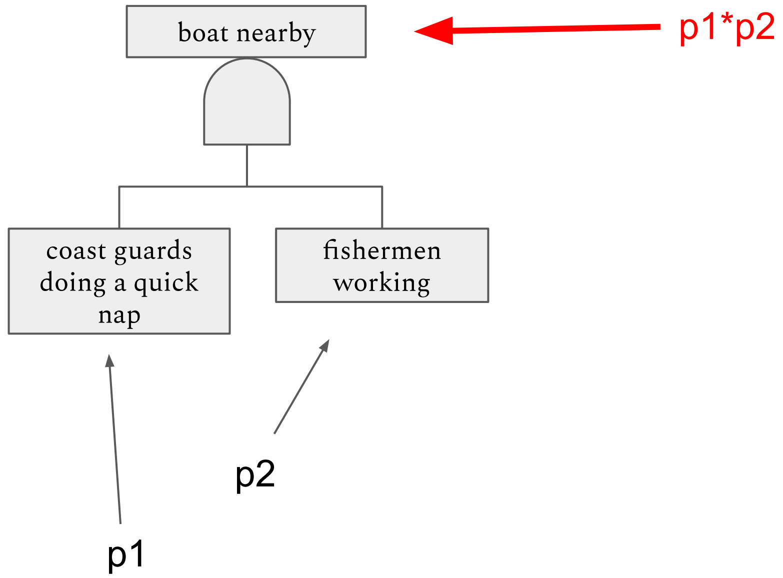
boat nearby
⁉️
coast guards doing a quick nap
fishermen working
p1
p2

AND gate



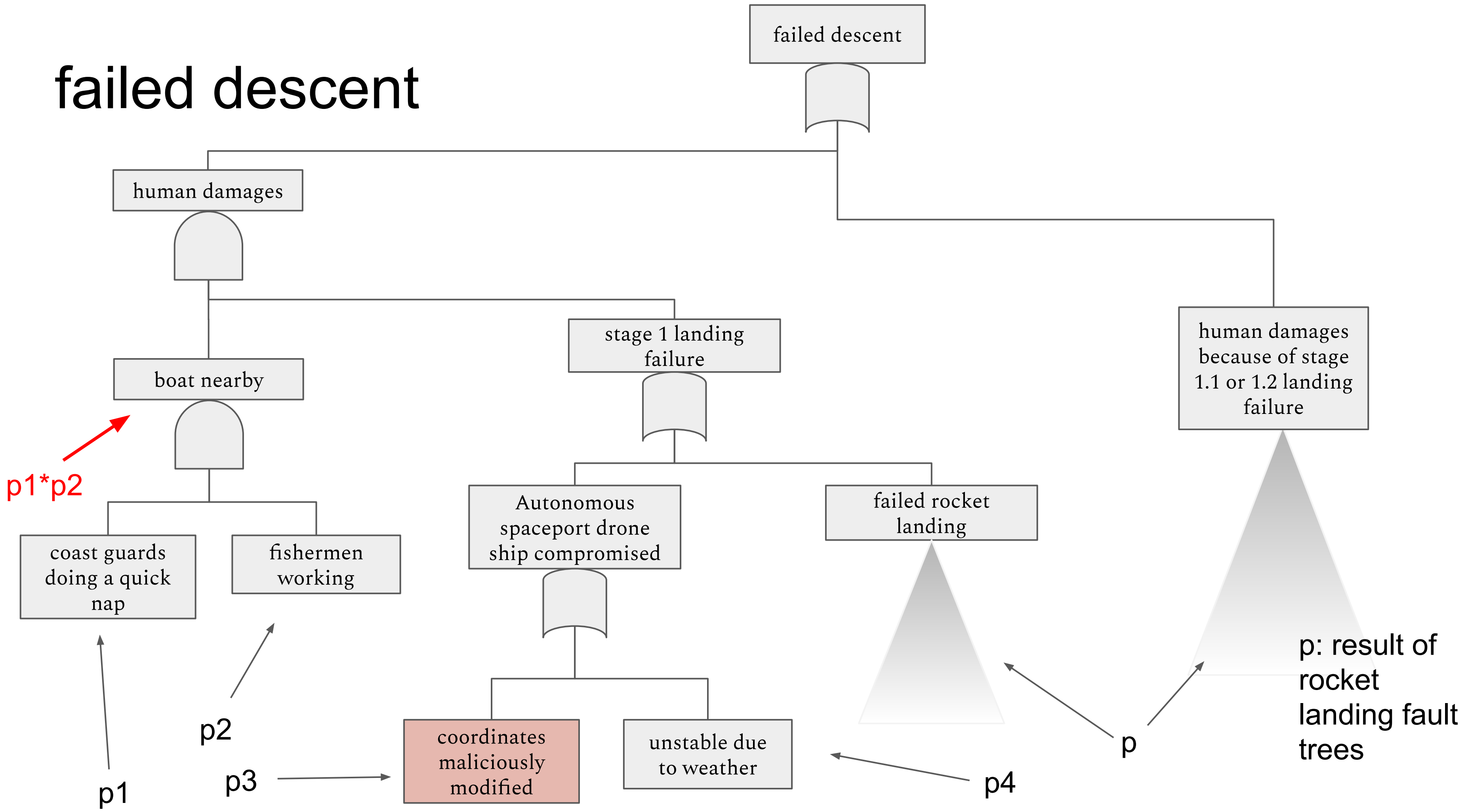
- Both events have to occur
- apparently independent events

AND gate

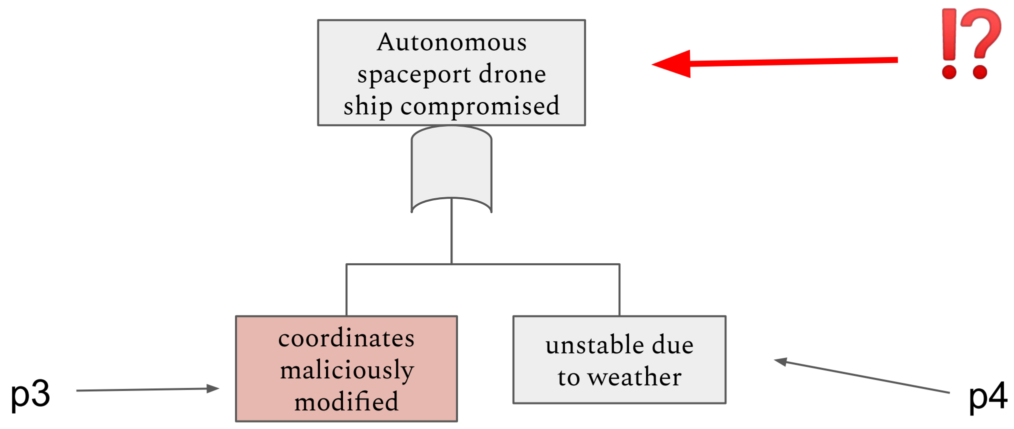


- Both events have to occur
- apparently independent events
- *P(boat nearby)=p1*p2*

failed descent

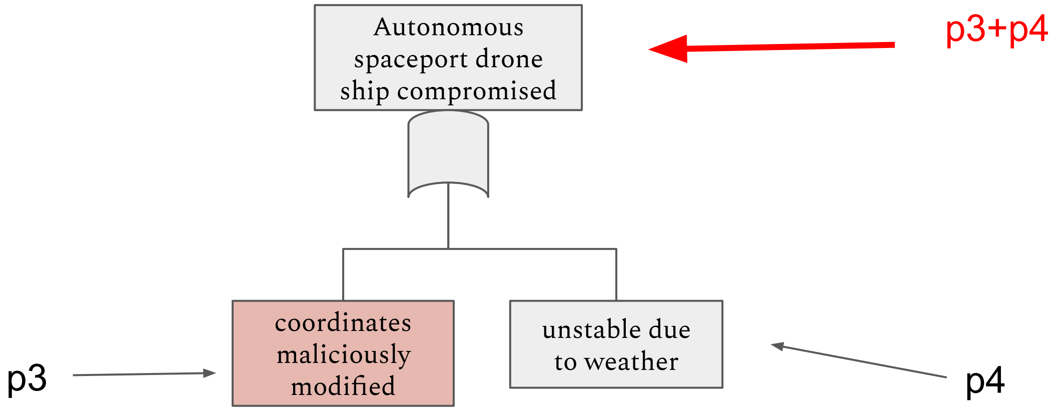

failed descent
human damages
boat nearby
p1*p2
coast guards doing a quick nap
fishermen working
p1
p2
stage 1 landing failure
Autonomous spaceport drone ship compromised
coordinates maliciously modified
unstable due to weather
p3
p4
failed rocket landing
human damages because of stage 1.1 or 1.2 landing failure
p
p: result of rocket landing fault trees

OR gate


⁉️
Autonomous spaceport drone ship compromised
coordinates maliciously modified
unstable due to weather
p3
p4

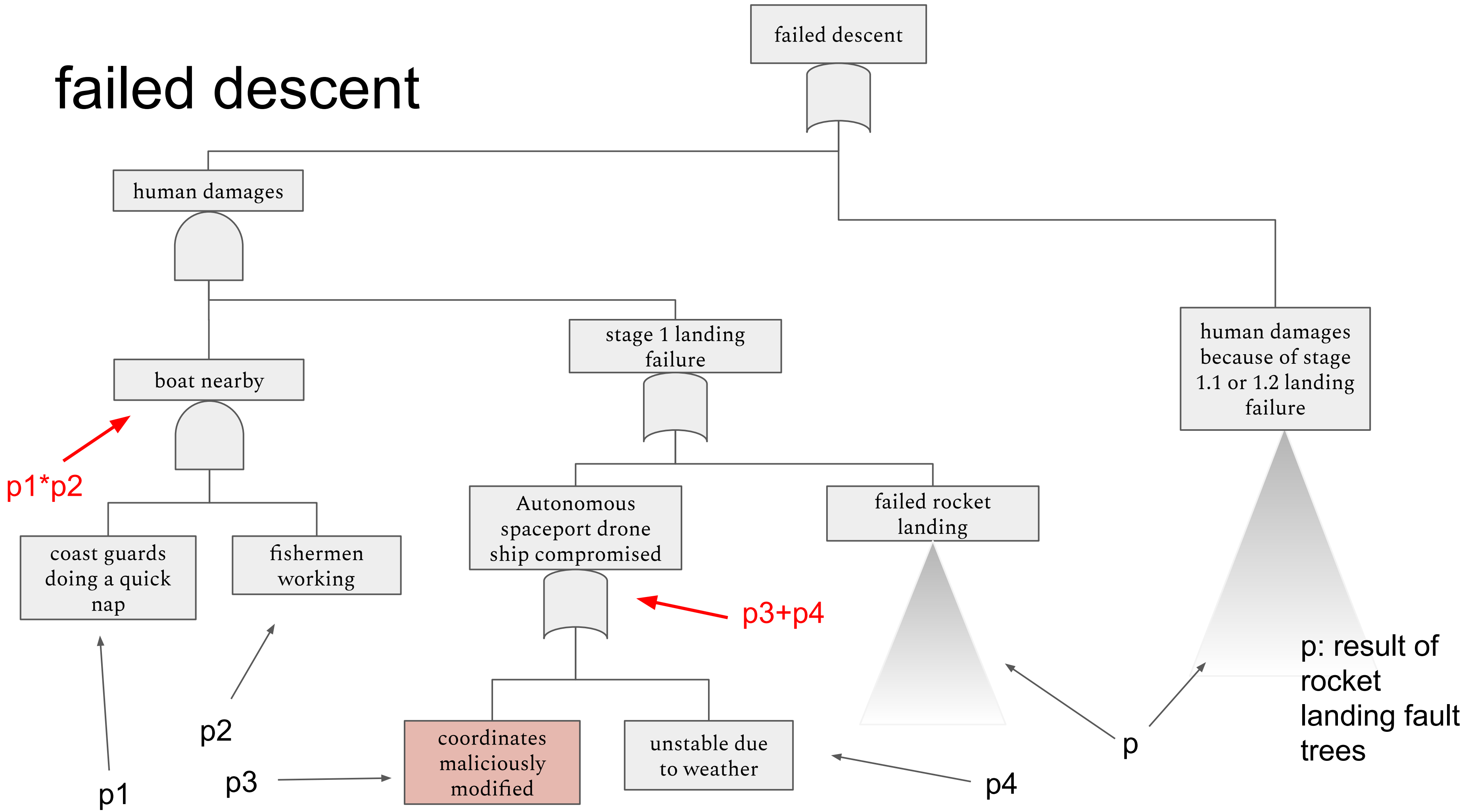
- at least one event has to occur
- apparently independent events

OR gate

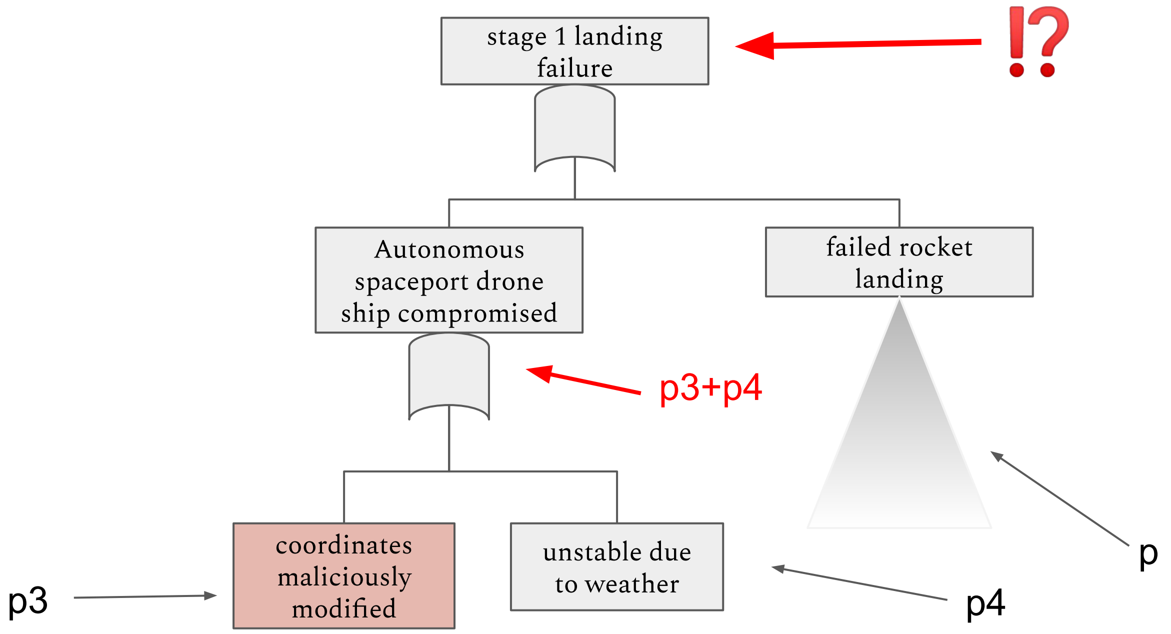

Autonomous spaceport drone ship compromised
p3+p4
coordinates maliciously modified
unstable due to weather
p3
p4

- at least one event has to occur
- apparently independent events
- *P(autonomous spaceport compromised)=p3+p4*

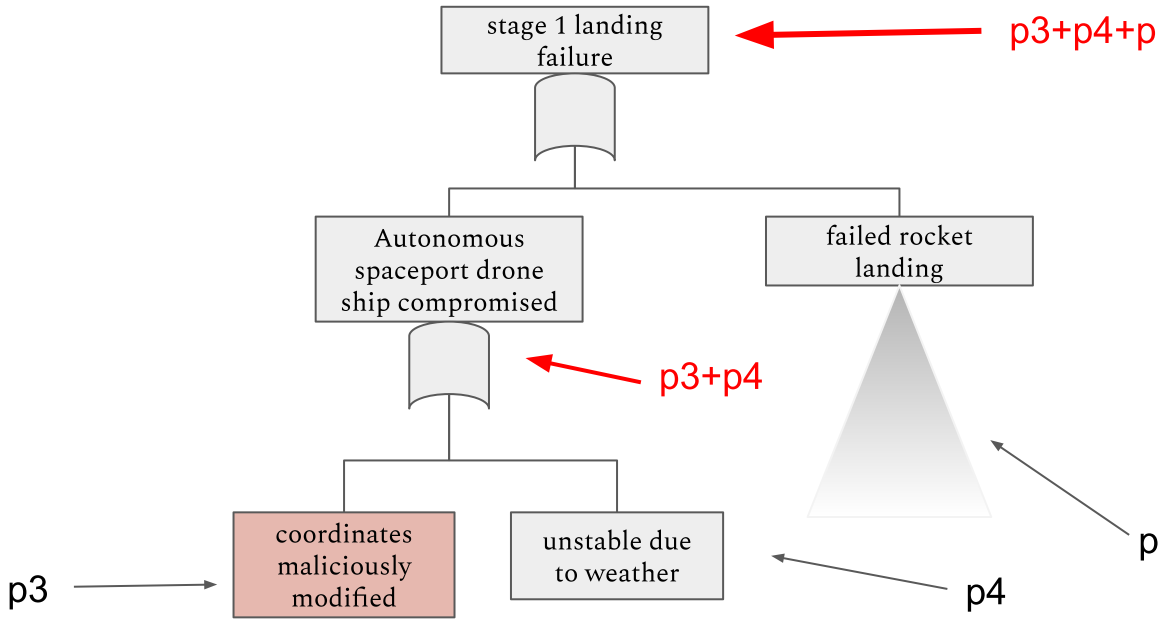
failed descent


failed descent
human damages
boat nearby
coast guards doing a quick nap
fishermen working
stage 1 landing failure
Autonomous spaceport drone ship compromised
coordinates maliciously modified
unstable due to weather
failed rocket landing
human damages because of stage 1.1 or 1.2 landing failure
p1*p2
p3+p4
p1
p2
p3
p4
p
p: result of rocket landing fault trees

OR gate

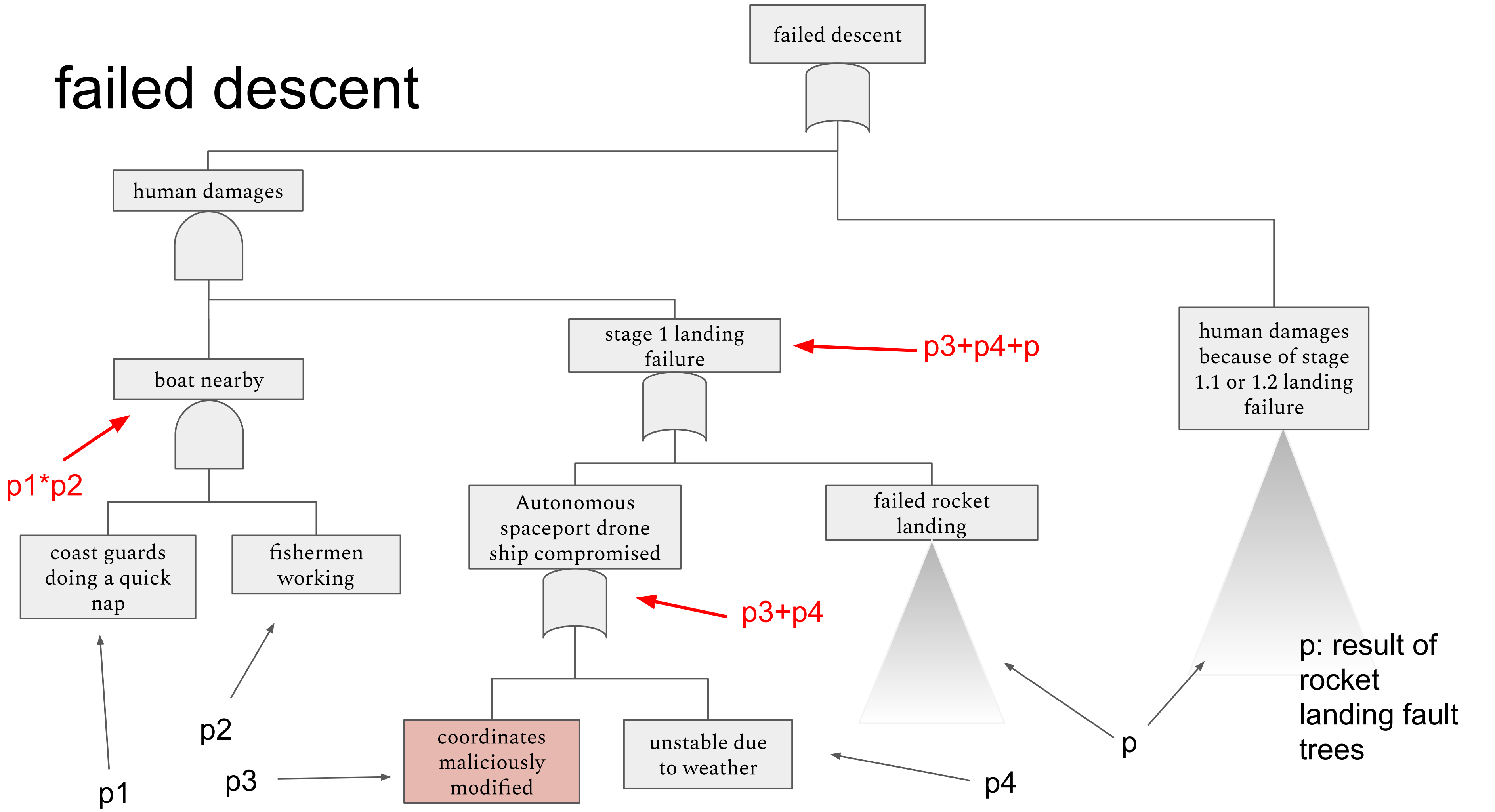

stage 1 landing failure
⁉️
Autonomous spaceport drone ship compromised
failed rocket landing
p3+p4
p
coordinates maliciously modified
unstable due to weather
p3
p4

OR gate

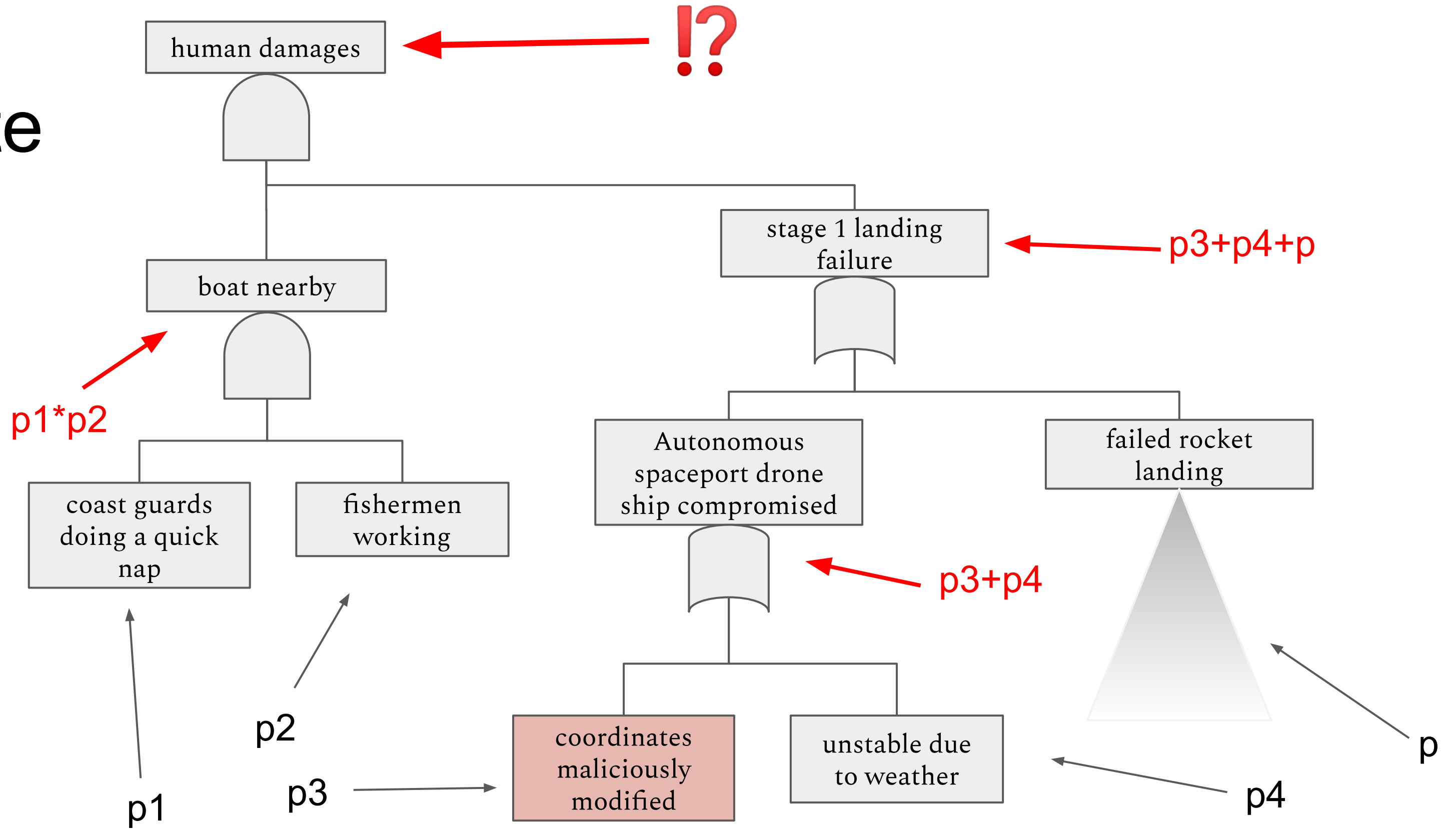


- *P(stage 1 landing failure)=p3+p4+p*

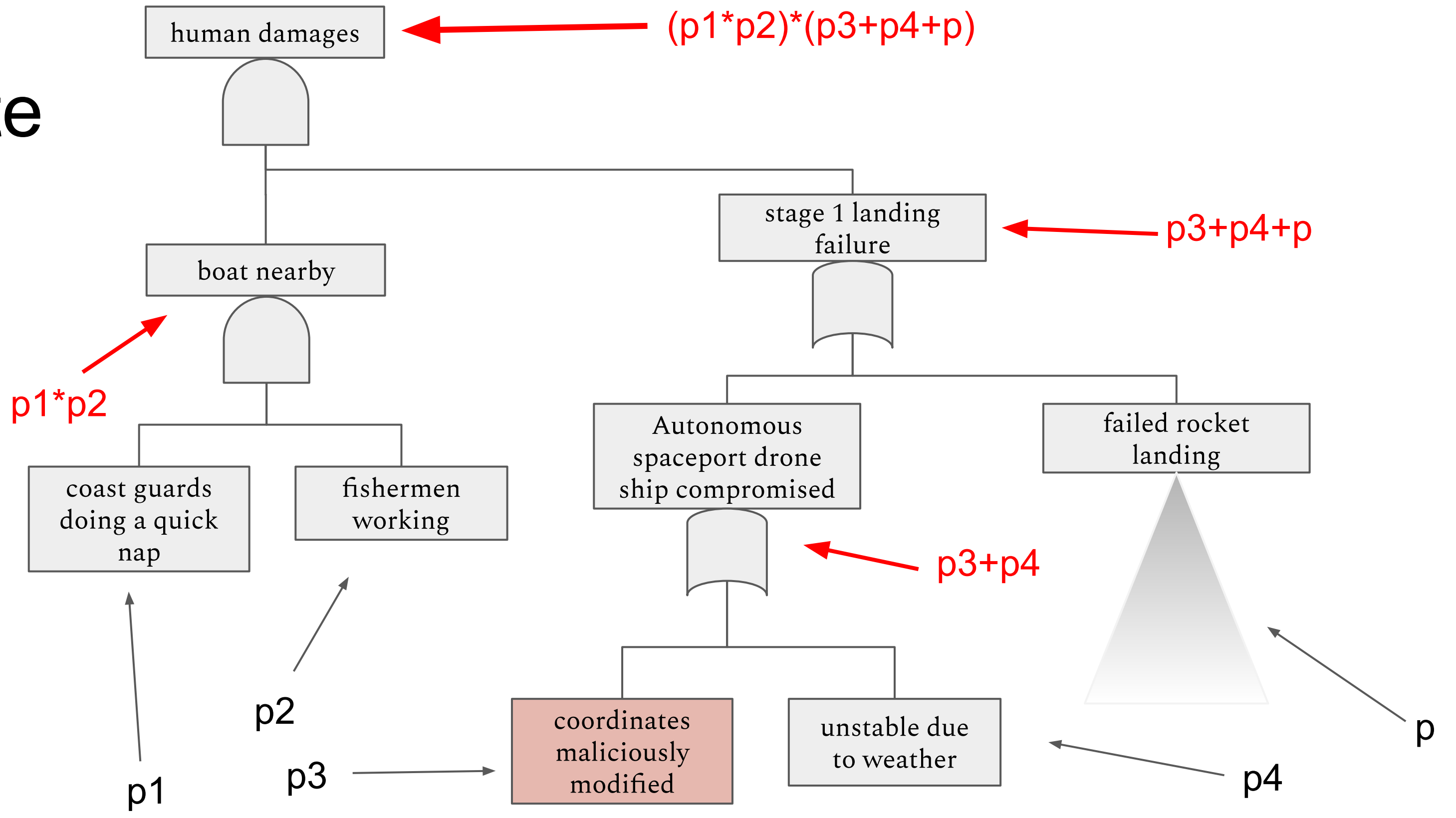
failed descent


failed descent
human damages
human damages because of stage 1.1 or 1.2 landing failure
stage 1 landing failure
p3+p4+p
boat nearby
p1*p2
Autonomous spaceport drone ship compromised
failed rocket landing
coast guards doing a quick nap
fishermen working
p3+p4
p: result of rocket landing fault trees
p2
coordinates maliciously modified
unstable due to weather
p
p1
p3
p4

AND gate

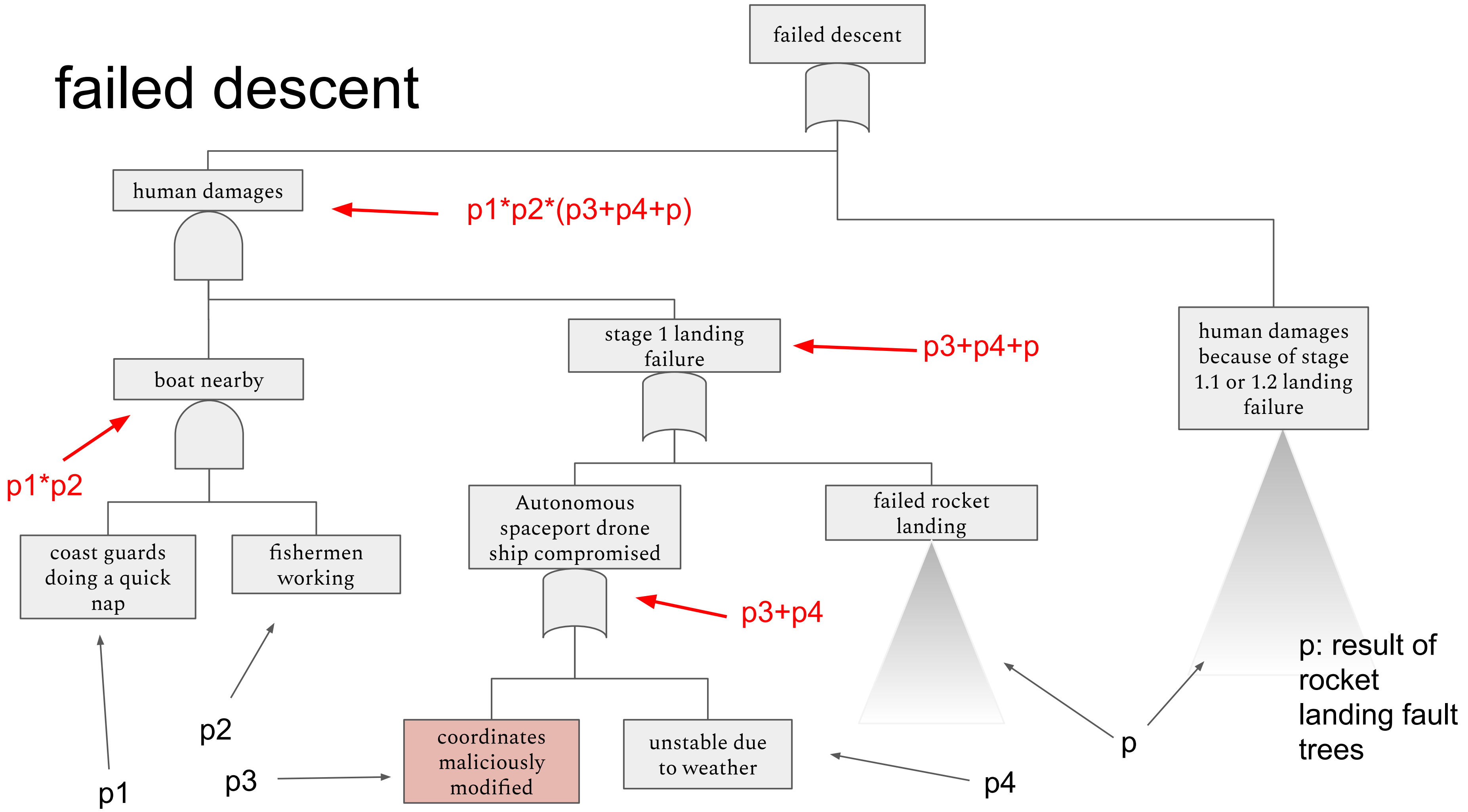

human damages
⁉️
stage 1 landing failure
p3+p4+p
boat nearby
p1*p2
Autonomous spaceport drone ship compromised
failed rocket landing
p3+p4
coast guards doing a quick nap
fishermen working
p2
p1
p3
coordinates maliciously modified
unstable due to weather
p4
p

AND gate

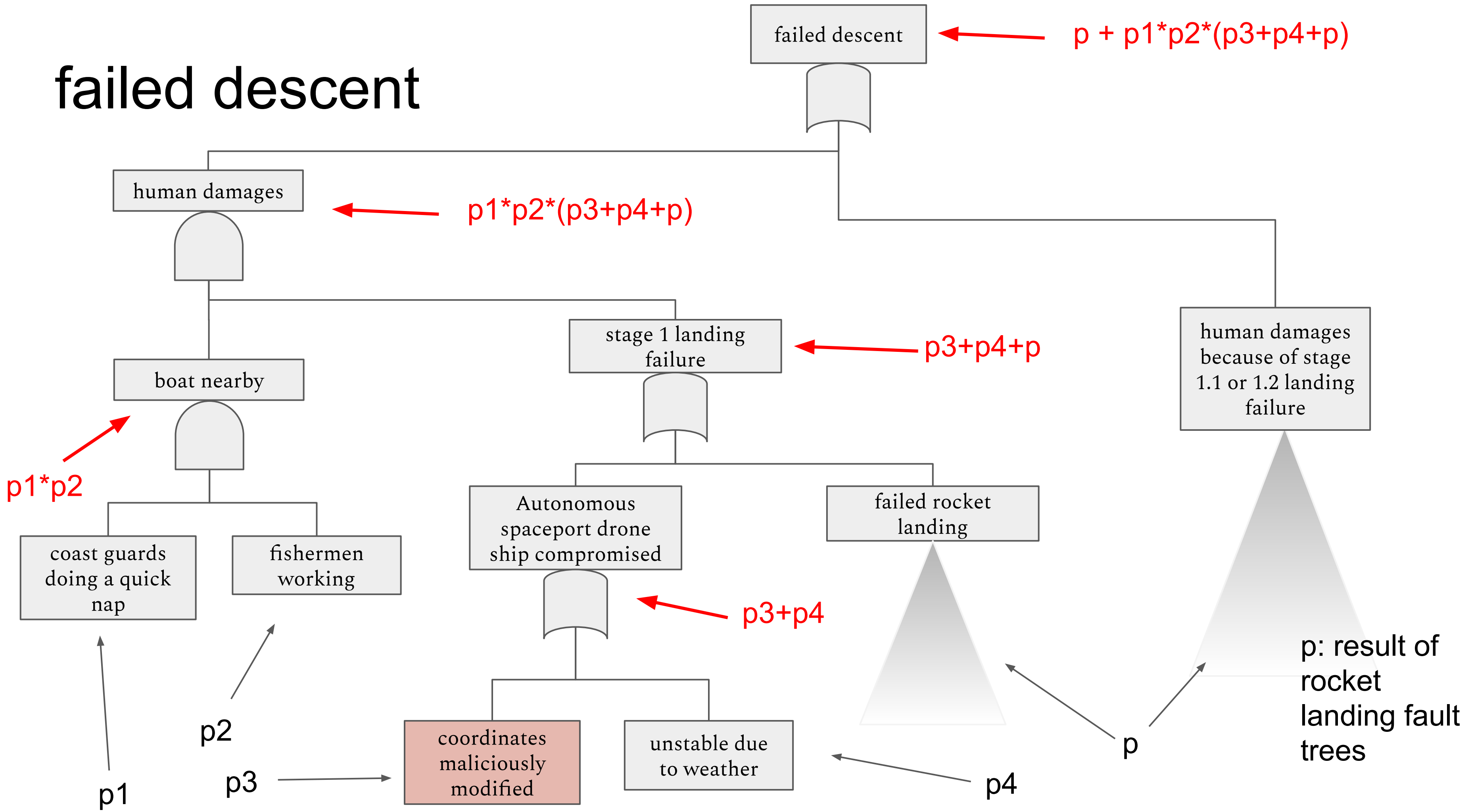


- *P(human damages)=(p1*p2)*(p3+p4+p)*

failed descent


failed descent
human damages
p1*p2*(p3+p4+p)
stage 1 landing failure
p3+p4+p
human damages because of stage 1.1 or 1.2 landing failure
boat nearby
p1*p2
coast guards doing a quick nap
fishermen working
Autonomous spaceport drone ship compromised
failed rocket landing
p3+p4
coordinates maliciously modified
unstable due to weather
p1
p2
p3
p4
p
p: result of rocket landing fault trees

failed descent


failed descent
p + p1*p2*(p3+p4+p)
human damages
p1*p2*(p3+p4+p)
boat nearby
p1*p2
stage 1 landing failure
p3+p4+p
human damages because of stage 1.1 or 1.2 landing failure
coast guards doing a quick nap
fishermen working
Autonomous spaceport drone ship compromised
failed rocket landing
p3+p4
coordinates maliciously modified
unstable due to weather
p1
p2
p3
p4
p
p: result of rocket landing fault trees

demo

- https://www.fault-tree-analysis-software.com/fault-tree-analysis

- create an account and log in: ismatbelval@gmail.com

  passwd: spaceinformatics

- download `failed descent.zip` from moodle

- fault tree → load from file

Attackers profiles

enable/disable attacks based on parameters

	no skill	medium skills	highly skilled
no budget			
medium budget			
high budget			

Attackers profiles

enable/disable attacks based on parameters

	no skill	medium skills	highly skilled
no budget			
medium budget			
high budget			

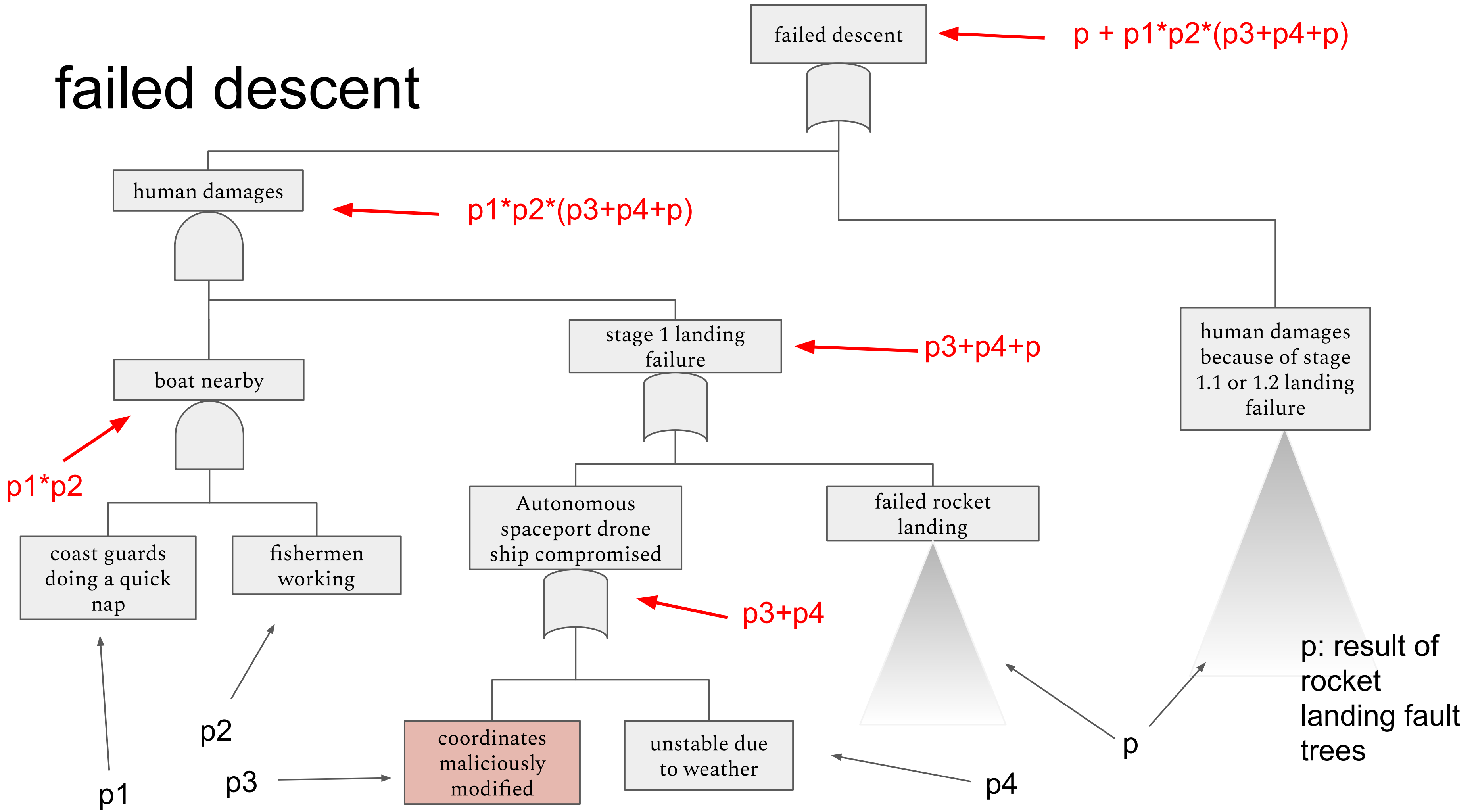
is the attack possible or not?
1 or 0
enable or disable

Attackers profiles

enable/disable attacks based on parameters

	no skill	medium skills	highly skilled
no budget			
medium budget			
high budget			Nation state

failed descent

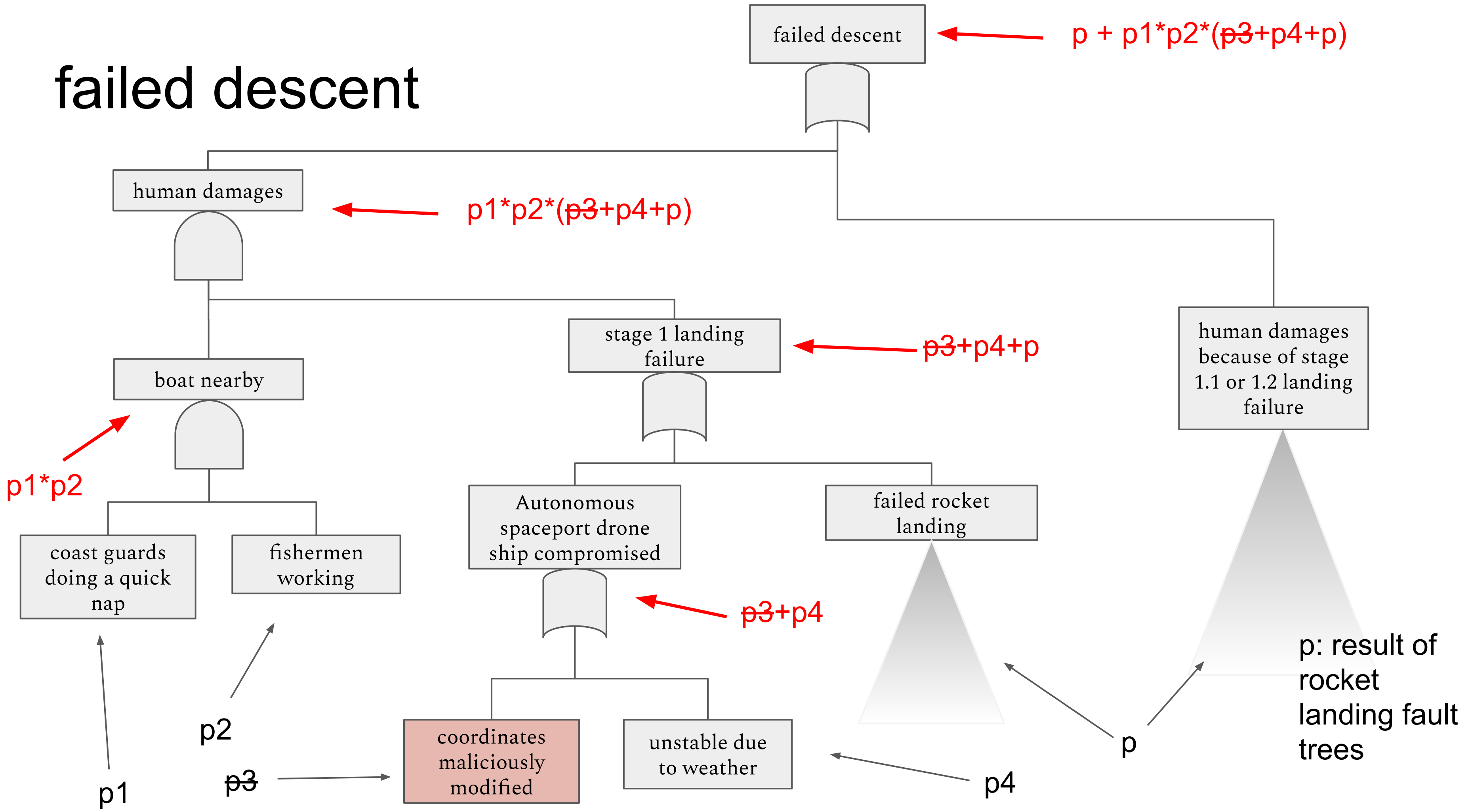

failed descent
p + p1*p2*(p3+p4+p)
human damages
p1*p2*(p3+p4+p)
stage 1 landing failure
p3+p4+p
human damages because of stage 1.1 or 1.2 landing failure
boat nearby
p1*p2
Autonomous spaceport drone ship compromised
failed rocket landing
coast guards doing a quick nap
fishermen working
p3+p4
coordinates maliciously modified
unstable due to weather
p2
p1
p3
p4
p
p: result of rocket landing fault trees

Attackers profiles

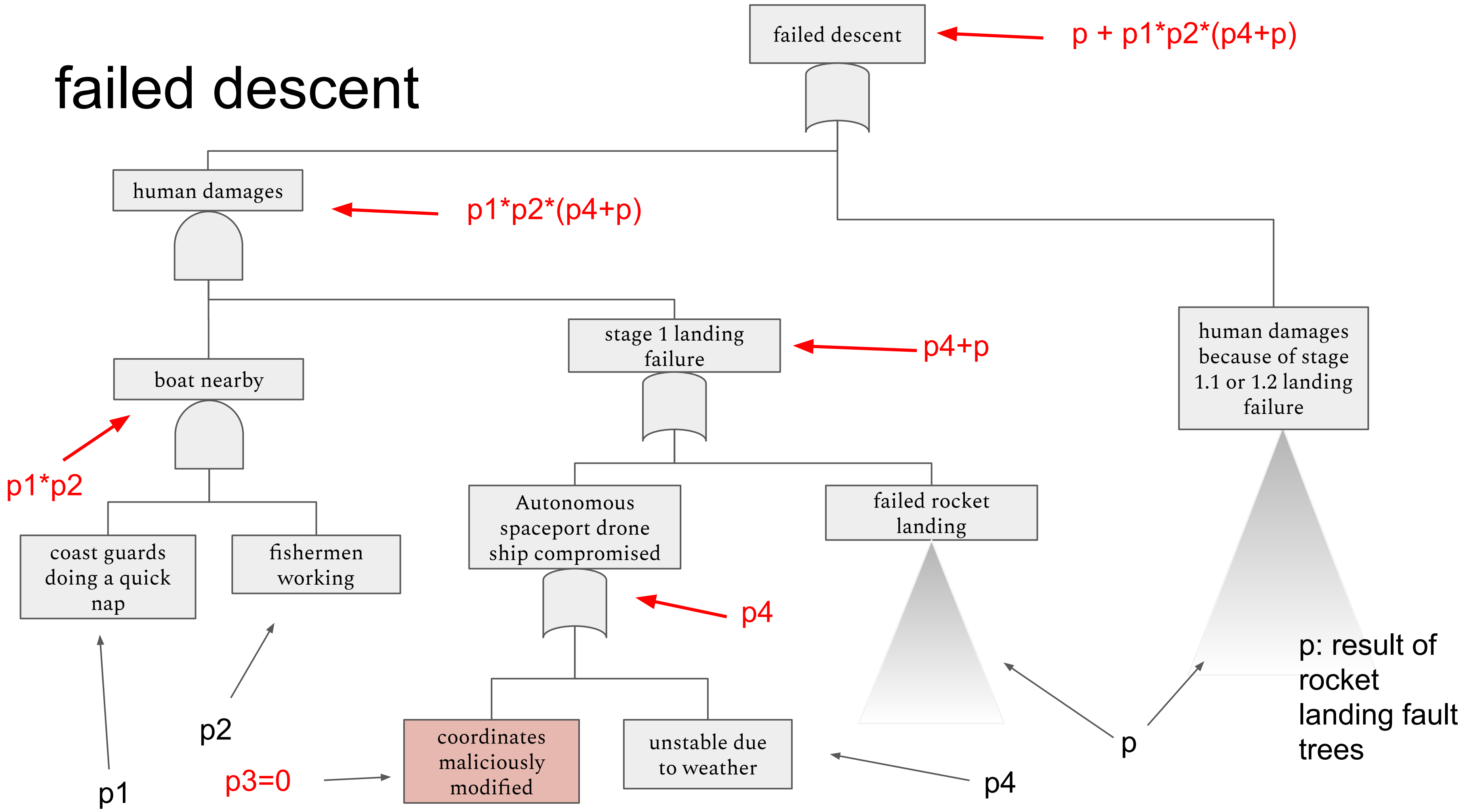
enable/disable attacks based on parameters

	no skill	medium skills	highly skilled
no budget	Newbie		
medium budget			
high budget			

failed descent



failed descent

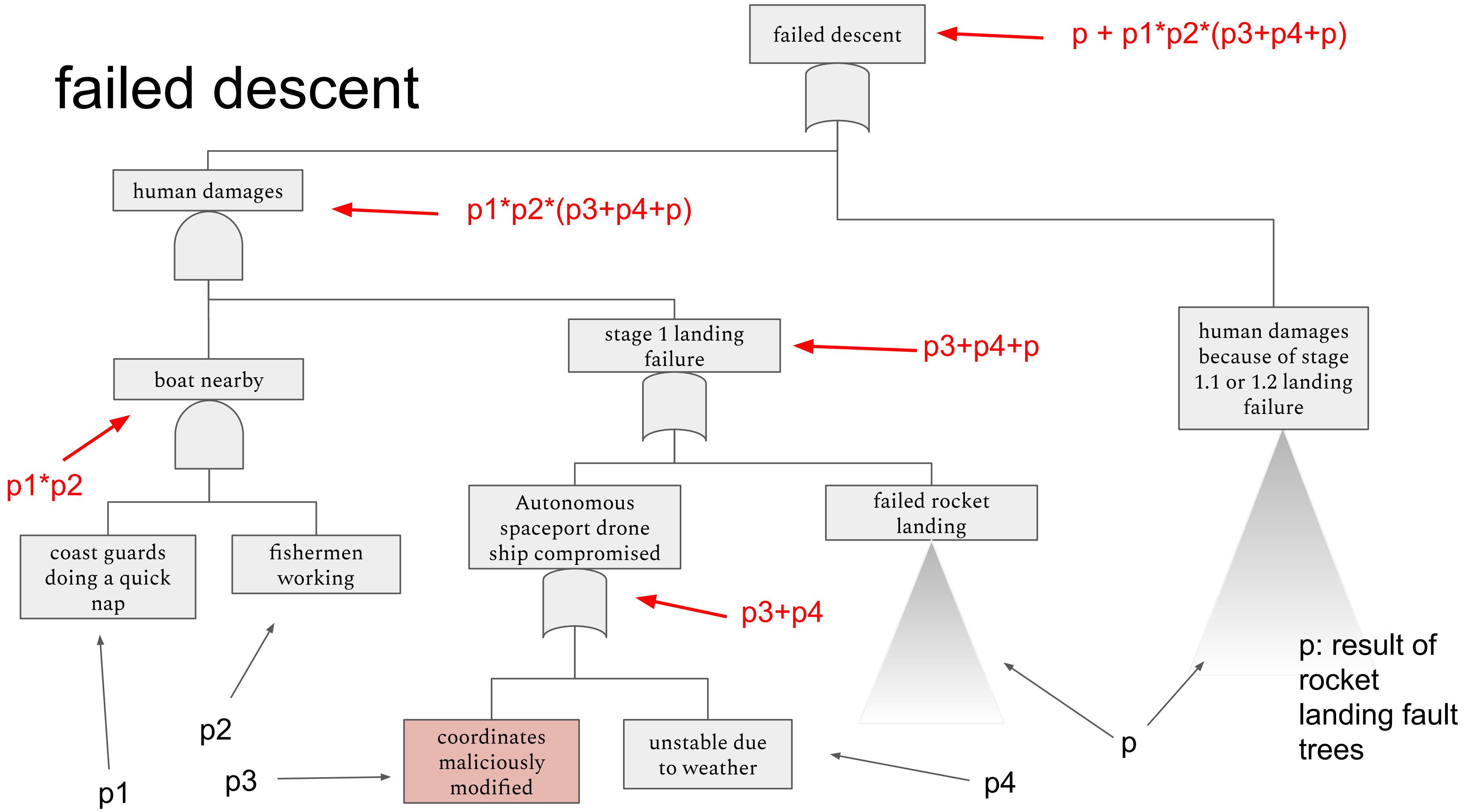

failed descent
p + p1*p2*(p4+p)
human damages
p1*p2*(p4+p)
stage 1 landing failure
p4+p
human damages because of stage 1.1 or 1.2 landing failure
boat nearby
p1*p2
Autonomous spaceport drone ship compromised
failed rocket landing
p4
coast guards doing a quick nap
fishermen working
coordinates maliciously modified
unstable due to weather
p2
p1
p3=0
p4
p
p: result of rocket landing fault trees

Attackers profiles

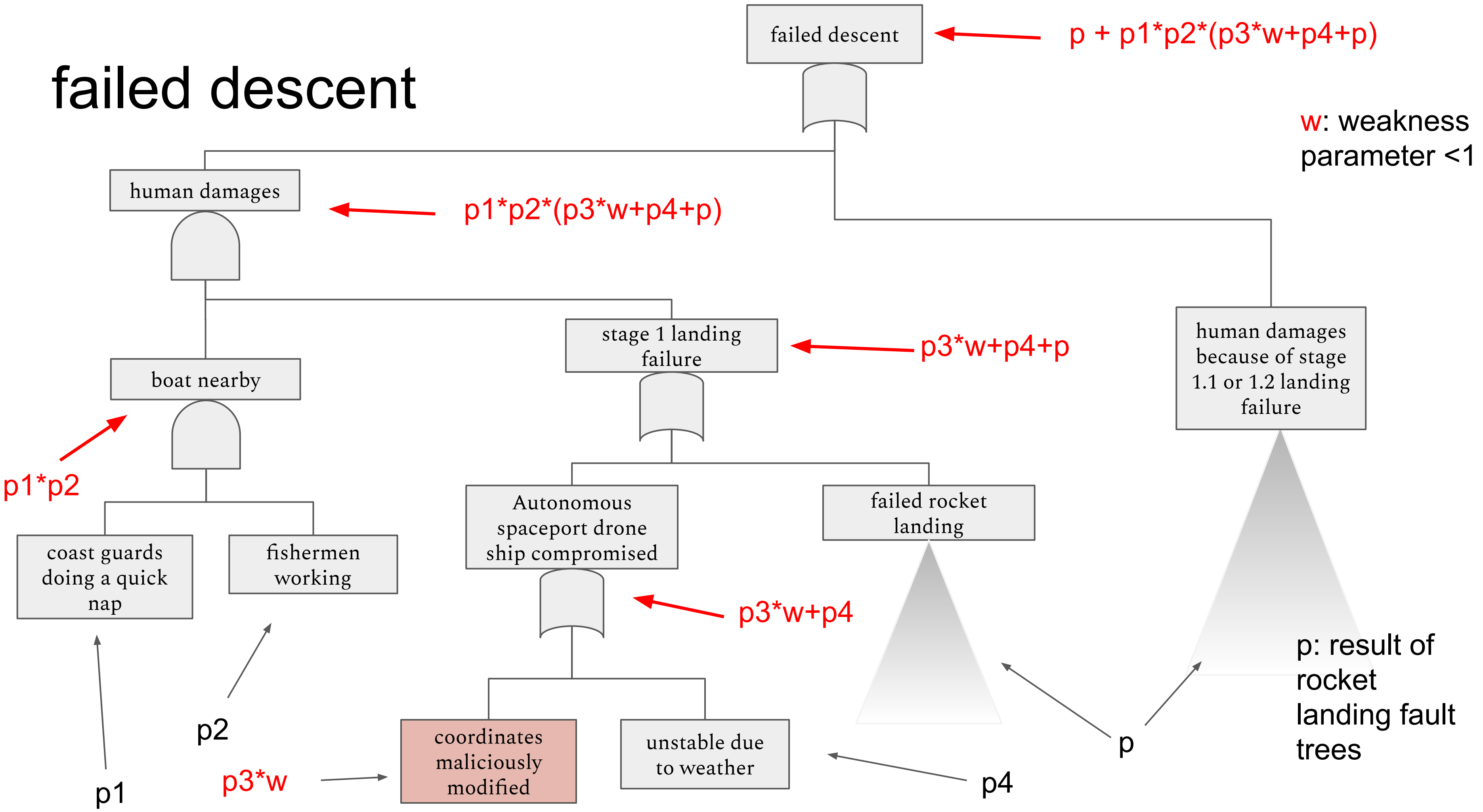
affects the probability of an attack to be successful

	no skill	medium skills	highly skilled
no budget		script kiddie	
medium budget			
high budget			

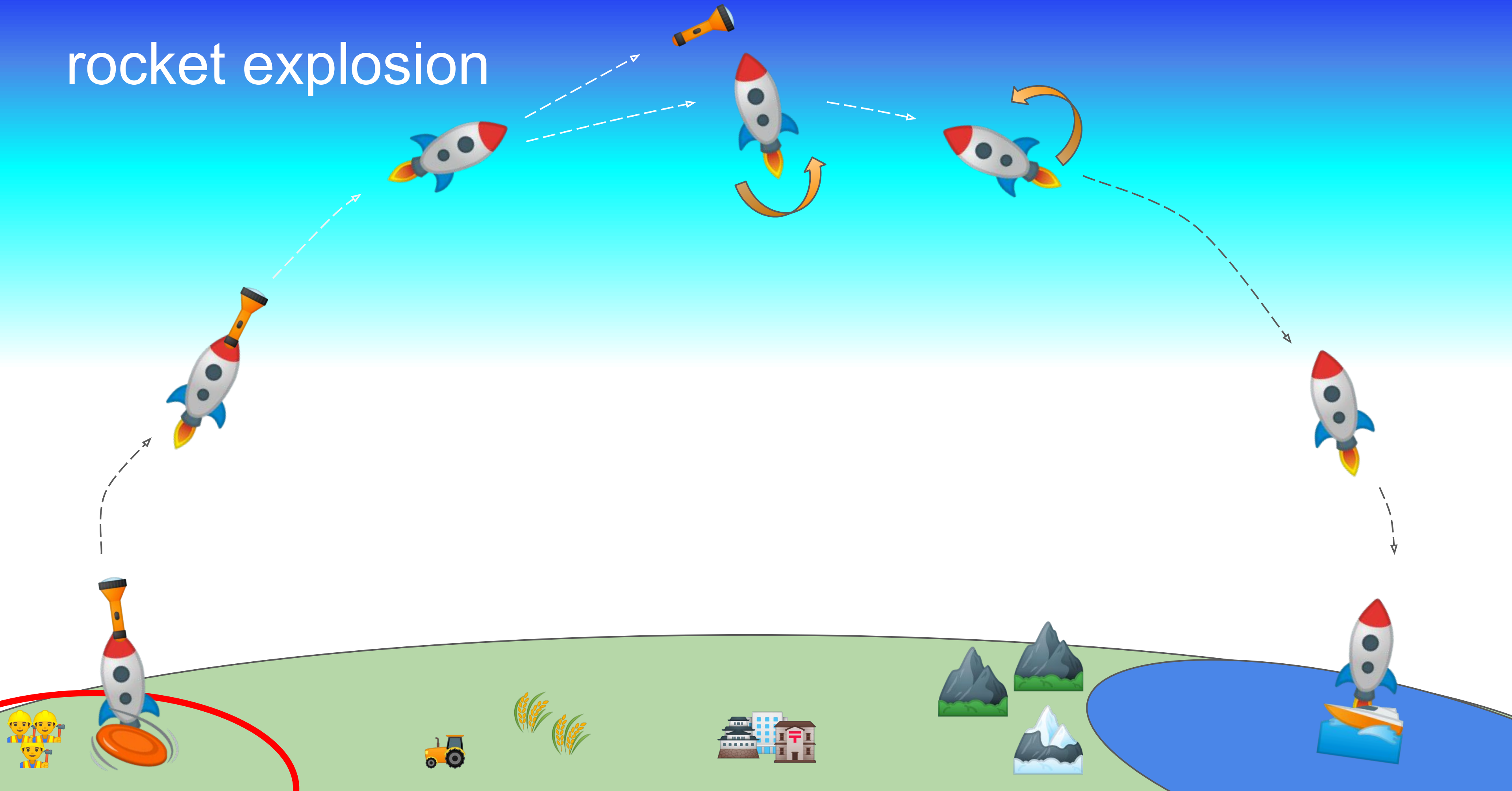
failed descent


failed descent
p + p1*p2*(p3+p4+p)
human damages
p1*p2*(p3+p4+p)
stage 1 landing failure
p3+p4+p
human damages because of stage 1.1 or 1.2 landing failure
boat nearby
p1*p2
Autonomous spaceport drone ship compromised
failed rocket landing
p3+p4
coast guards doing a quick nap
fishermen working
coordinates maliciously modified
unstable due to weather
p1
p2
p3
p4
p
p: result of rocket landing fault trees

failed descent



rocket explosion



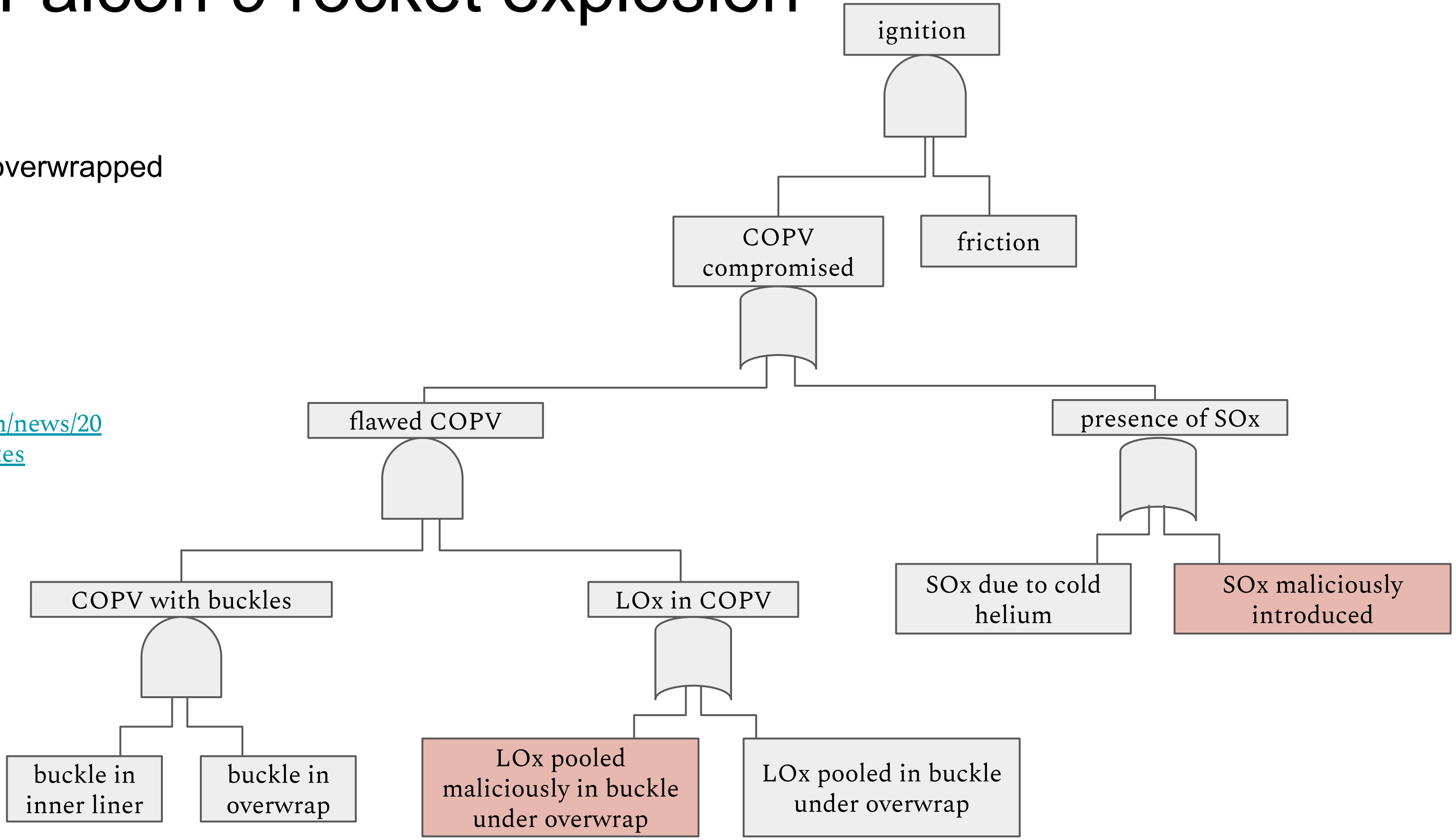
SpaceX Falcon 9 rocket explosion

COPV: composite overwrapped pressure vessel

SOx: solid oxygen

LOx: liquid oxygen

https://www.spacex.com/news/2016/09/01/anomaly-updates



demo

- https://www.fault-tree-analysis-software.com/fault-tree-analysis

- create an account and log in: ismatbelval@gmail.com

  passwd: spaceinformatics

- download `rocket explosion.zip` from moodle

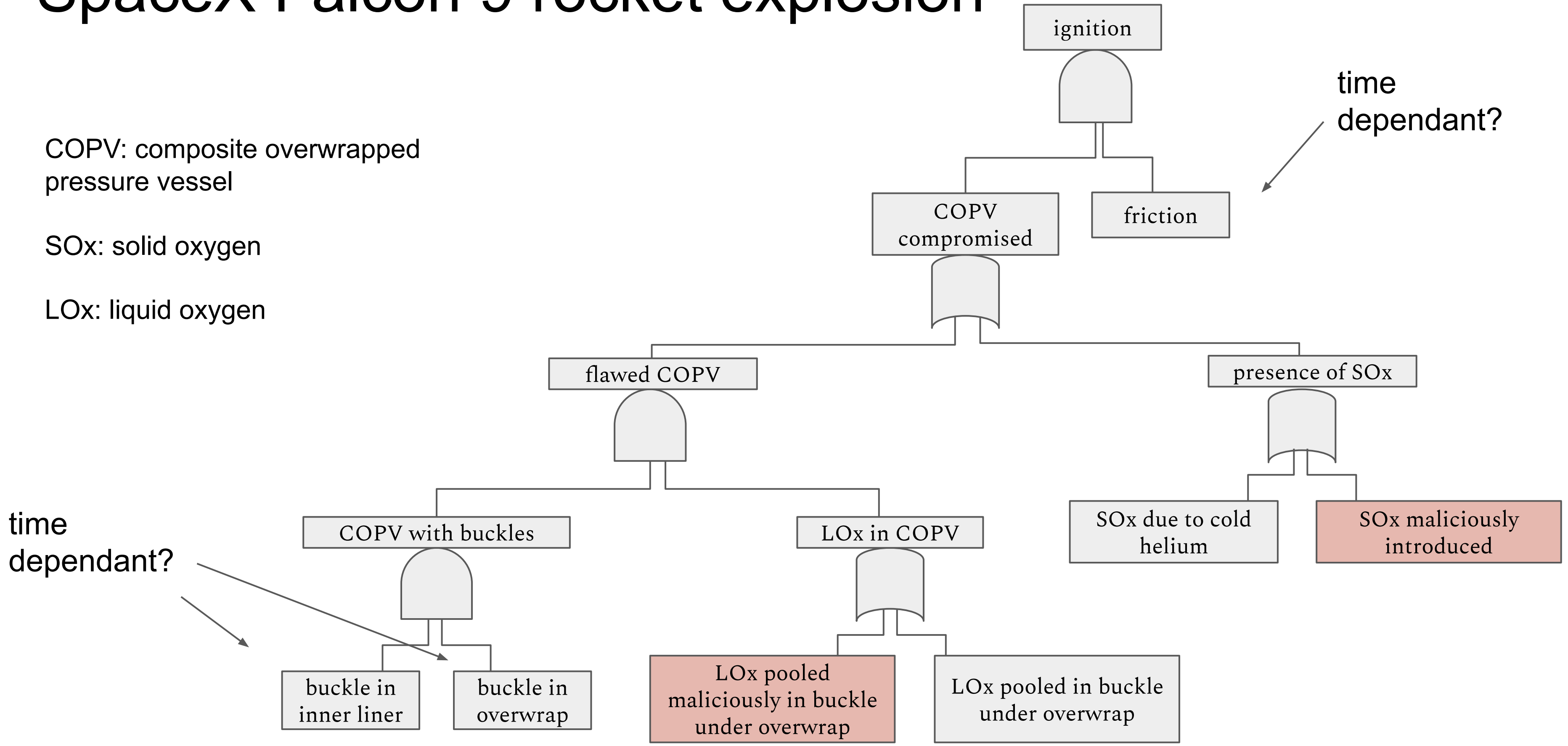
- fault tree → load from file

SpaceX Falcon 9 rocket explosion

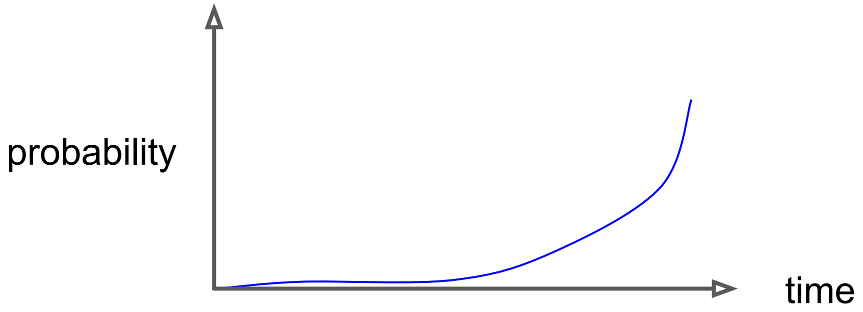
COPV: composite overwrapped pressure vessel

SOx: solid oxygen

LOx: liquid oxygen

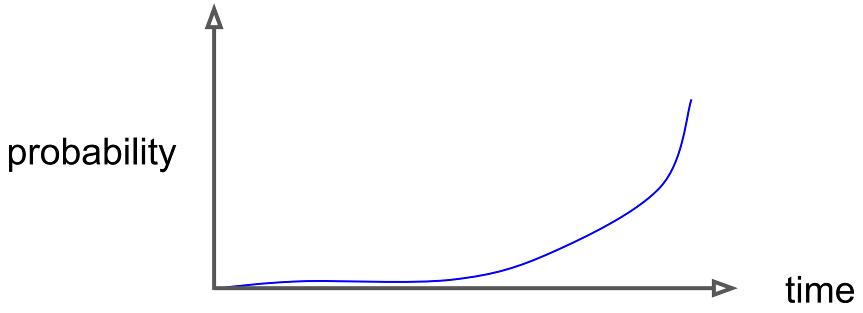


time dependant probabilistic events

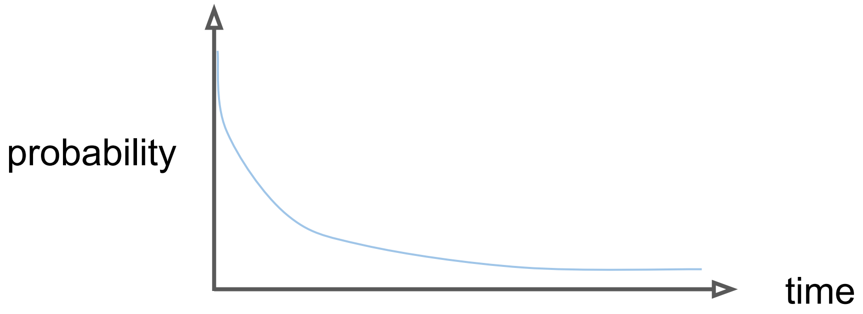


friction

time dependant probabilistic events



friction



buckle in inner liner

events modifications

- https://www.fault-tree-analysis-software.com/fault-tree-analysis

- fault tree → load from file
- select an event
- right click → edit

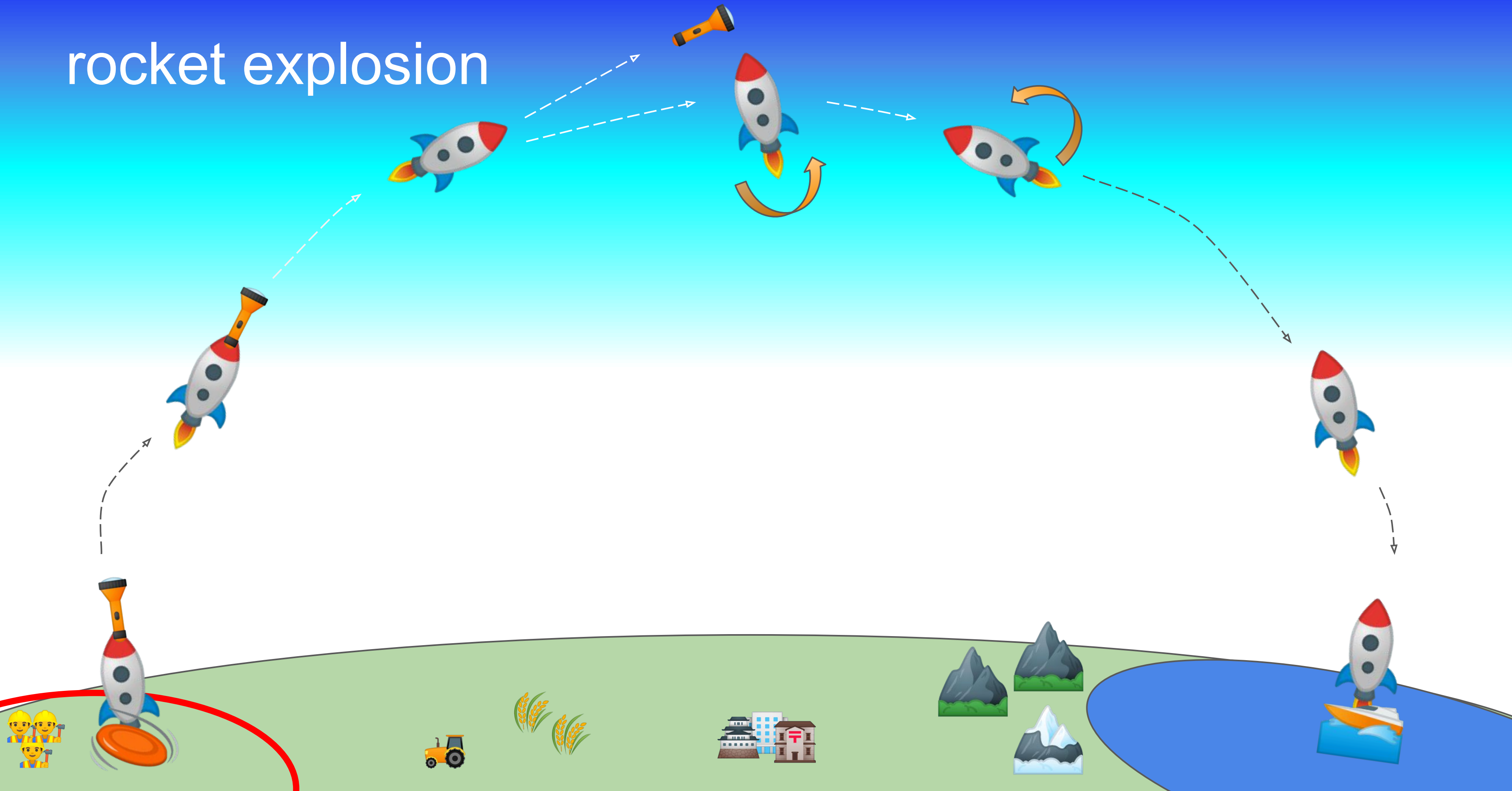
Attackers profiles

	no skill	medium skills	highly skilled
no budget			
medium budget			
high budget			

affects parameters defining the success of an attack
e.g. time, cost, damages

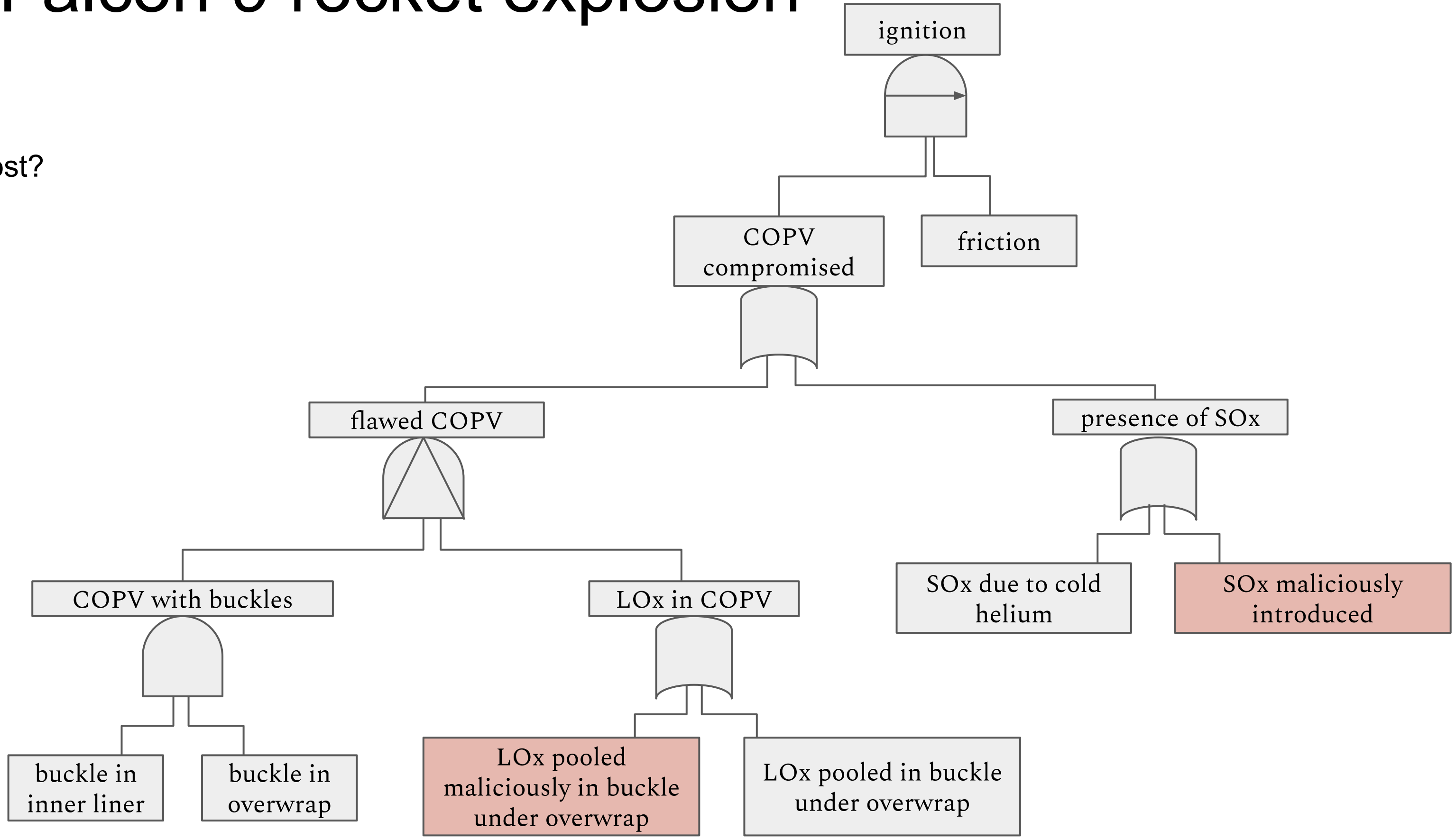
1. Fault tree analysis: probabilistic events
2. Fault tree analysis: costs and damages computation?
3. group work

rocket explosion



SpaceX Falcon 9 rocket explosion

everything has a cost?



demo iFat

- http://ctit-vm1.ewi.utwente.nl/FT_analysis/

- download `rocket explosion.json` from moodle
- file → load file
- add costs to events with the left panel attributes
- compute the final cost in the right panel

Problems:

Few intuitive tools

- FTA software

few gates
many models for events

- iFat

beta version (probabilities not working?)
many gates
only one cost parameter

Problems:

More complete (and more complex) tools

- combine costs, and probabilities (Uppaal SMC + ATTop)
- combine probabilities and time (COMPASS)
- combine costs, damages and time (imitator + ATTop)
- ADtool, ATCalc, Attack Tree Evaluator…

Problems:

More complete (and more complex) tools

- combine costs, and probabilities (Uppaal SMC + ATTop)
- combine probabilities and time (COMPASS)
- combine costs, damages and time (imitator + ATTop)
- ADtool, ATCalc, Attack Tree Evaluator…

question: *can we combine costs, time, probabilities in the same tool, and perform optimization procedures?*

→ for the infrastructure: maximize the duration of the attack, while keeping the damages low
→ for the attacker: given an event with a low probability, minimize the duration of an attack while keeping the cost low

Related work

Rajesh Kumar, Mariëlle Stoelinga: Quantitative Security and Safety Analysis with Attack-Fault Trees. HASE 2017

Étienne André, Didier Lime, Mathias Ramparison, Mariëlle Stoelinga: Parametric Analyses of Attack-Fault Trees. ACSD 2019

Marlon Fraile, Margaret Ford, Olga Gadyatskaya, Rajesh Kumar, Mariëlle Stoelinga, Rolando Trujillo-Rasua: Using Attack-Defense Trees to Analyze Threats and Countermeasures in an ATM: A Case Study. PoEM 2016

https://www.buran.su/buranvssts-comparison.php

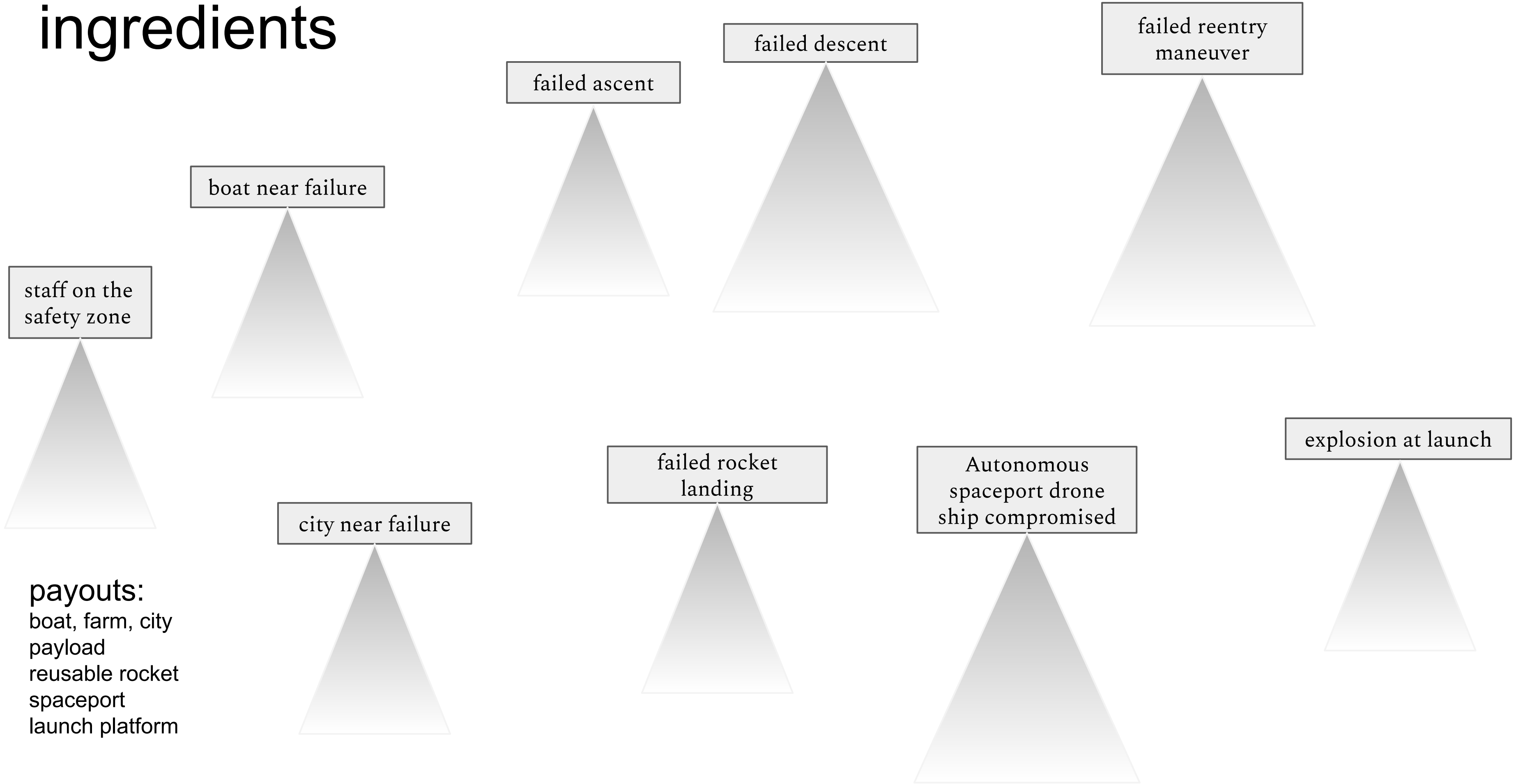
1. Fault tree analysis: probabilistic events
2. Fault tree analysis: costs and damages computation?

3. group work

open question:

how can we determine risk assessment, from fault trees and costs (to the organization, infrastructure, third party properties) caused by the failure of a fault tree?

ingredients



payouts:

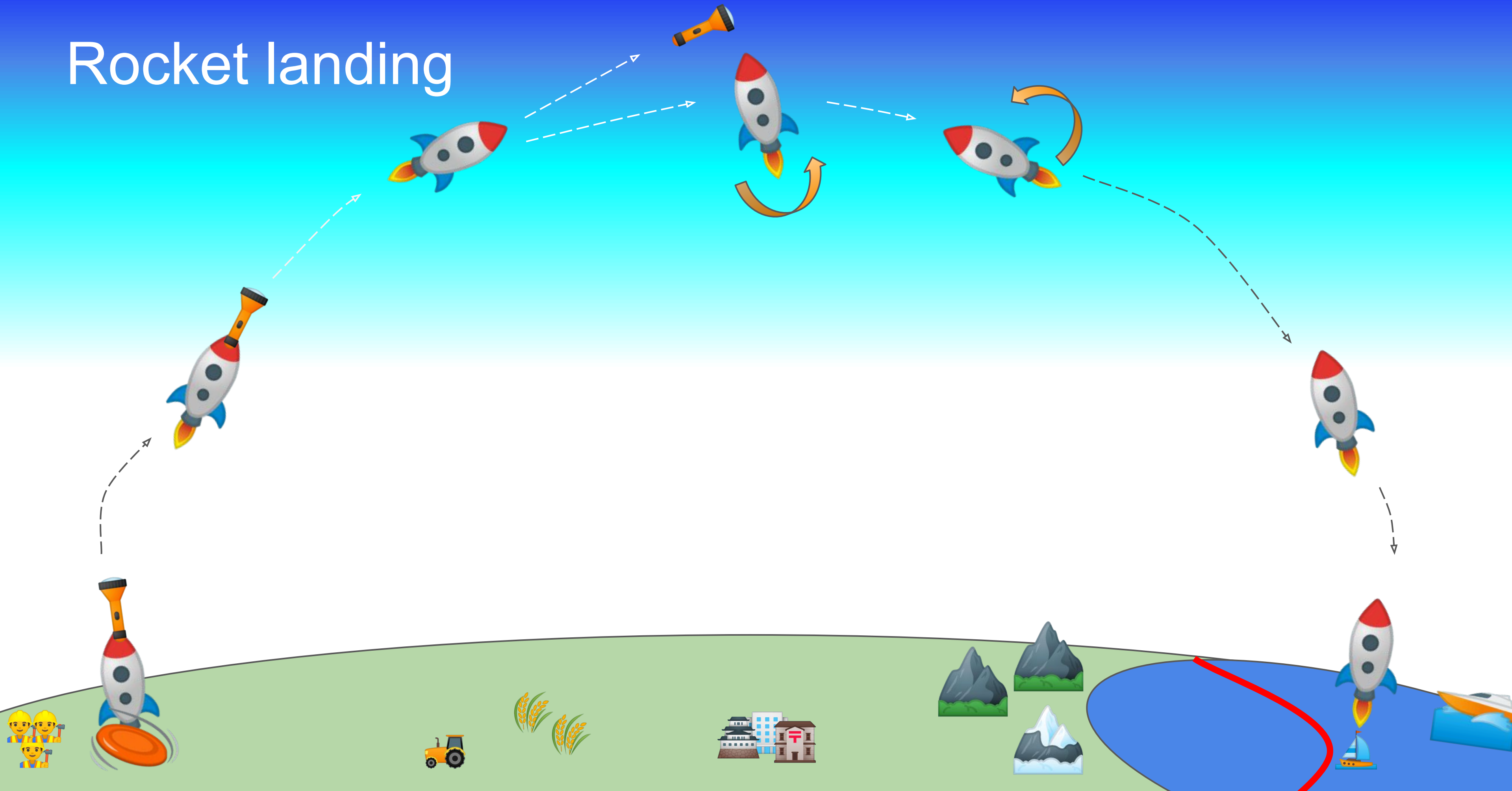
boat, farm, city
payload
reusable rocket
spaceport
launch platform

Attackers profiles

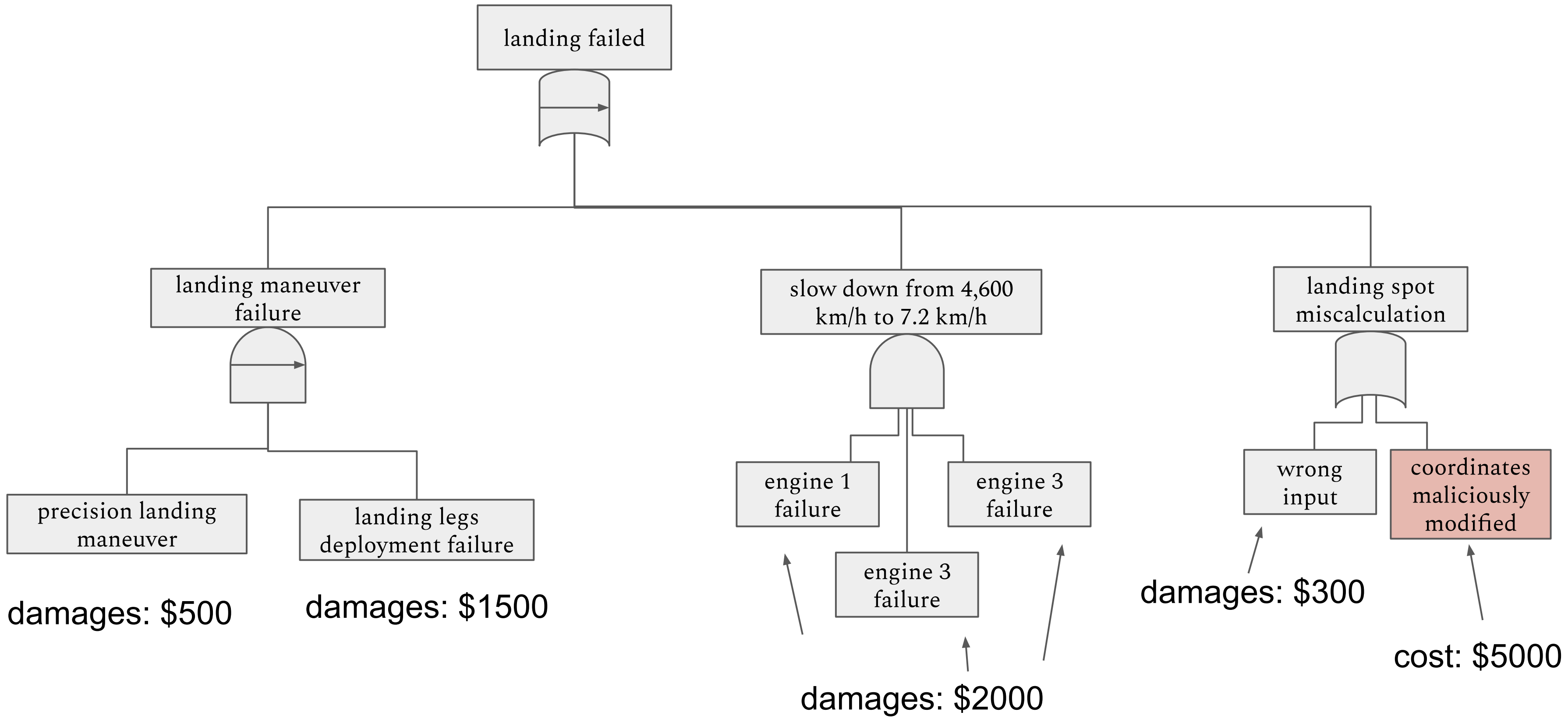
affects parameters defining the success of an attack e.g. time, cost, damages

	no skill	medium skills	highly skilled
no budget			
medium budget			
high budget			Nation state

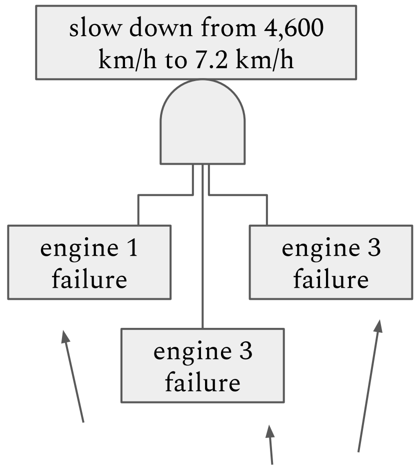
Rocket landing



Rocket landing

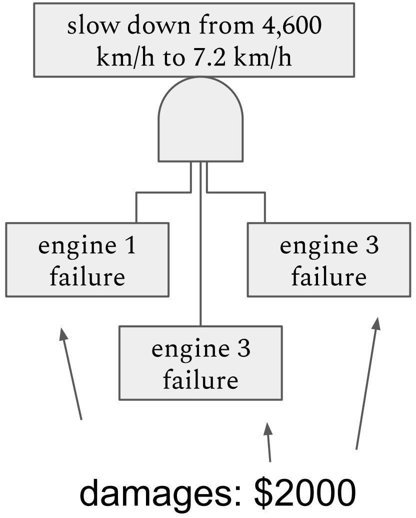


AND gate



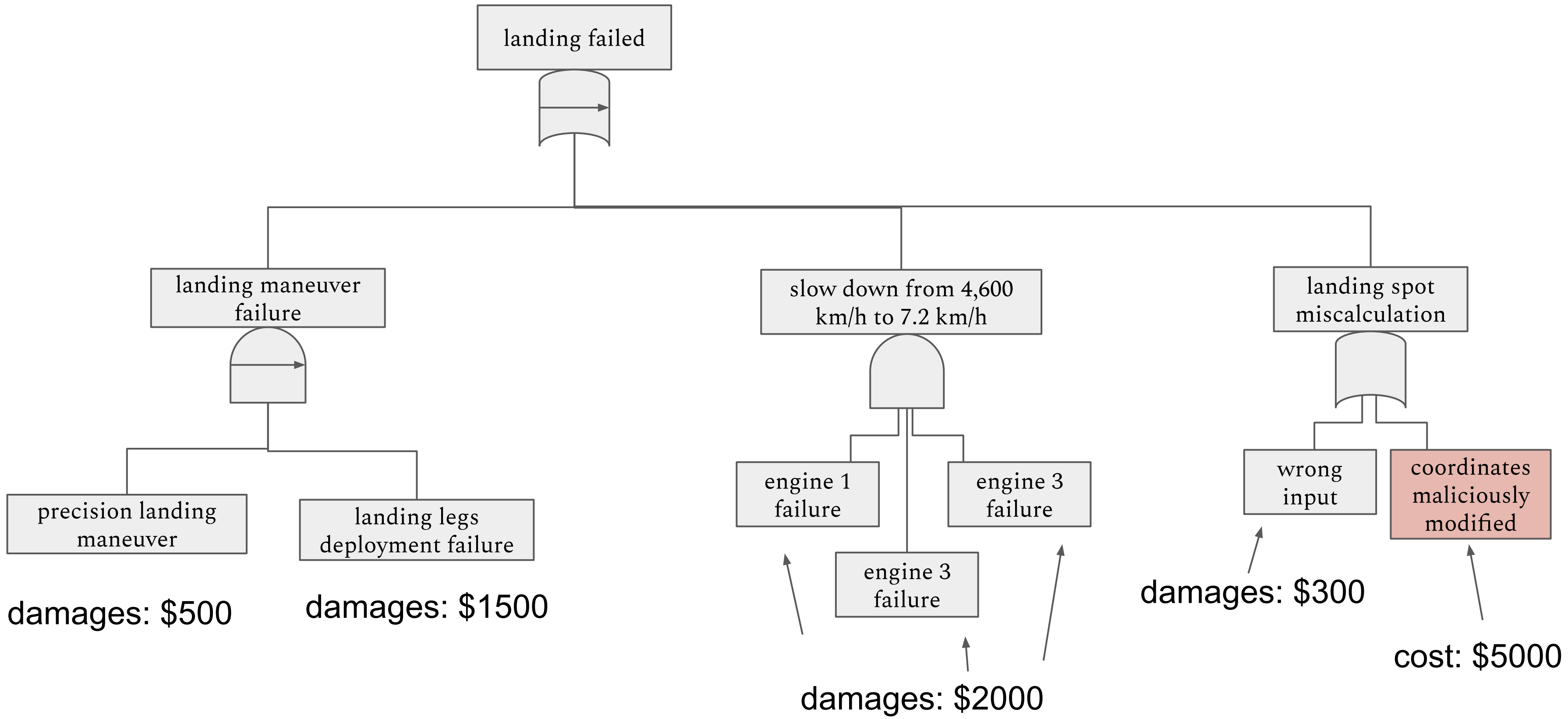
damages: $2000

AND gate

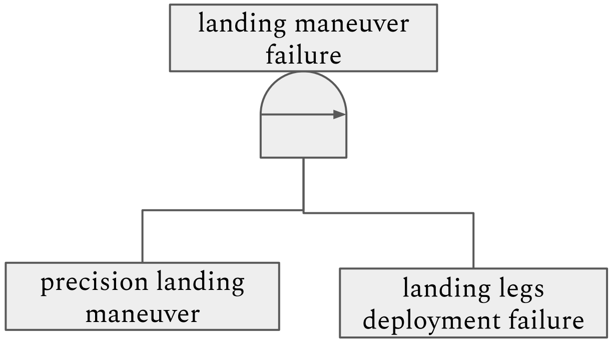


- *Damages(slow down)=$6000*

Rocket landing



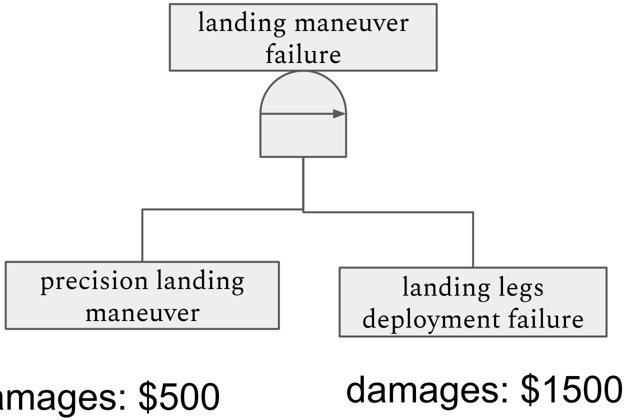
SAND gate



damages: $500

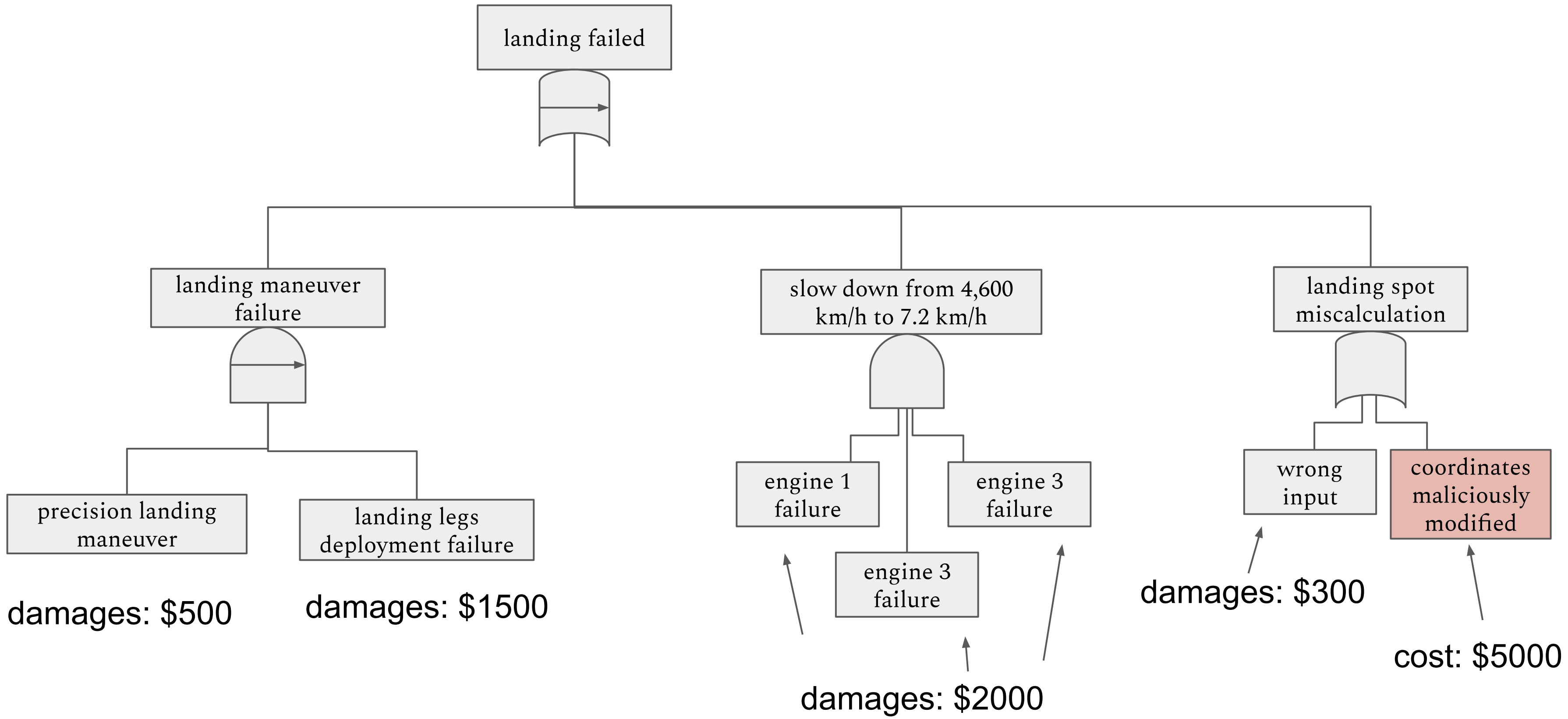
damages: $1500

SAND gate

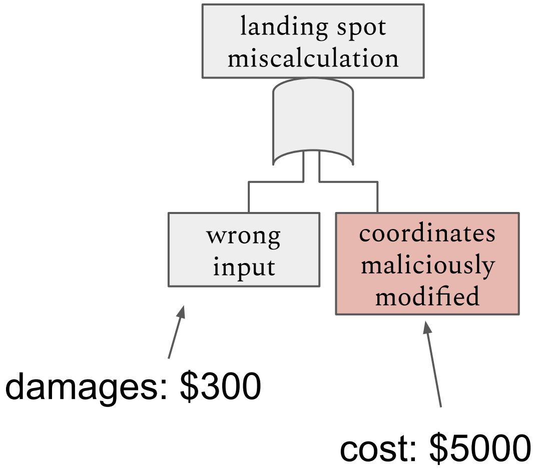


- *Damages(landing maneuver failure)=$2000*

Rocket landing

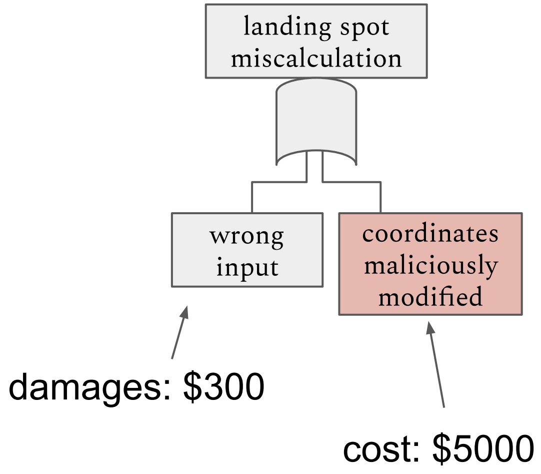


OR gate



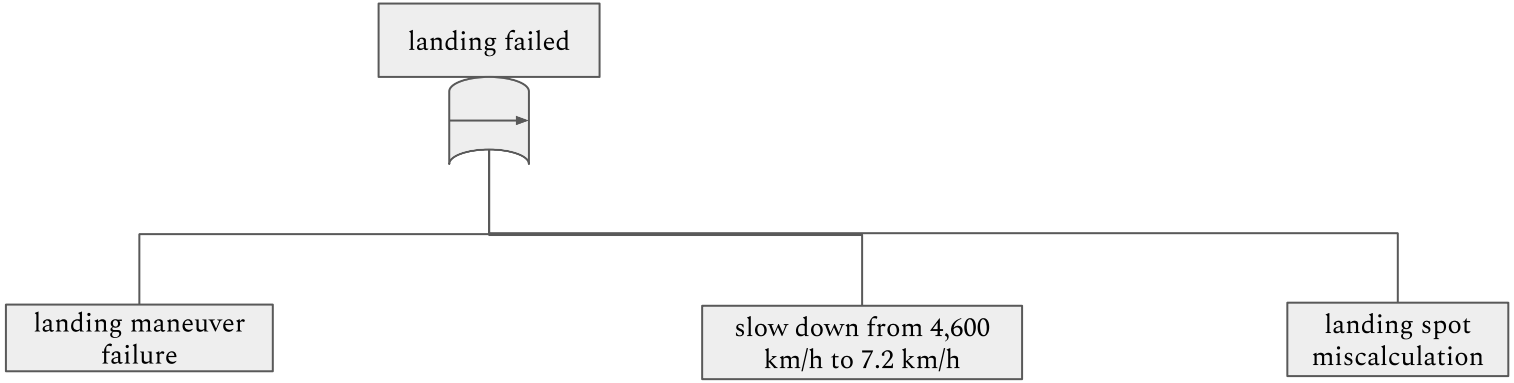
landing spot miscalculation
wrong input
coordinates maliciously modified
damages: $300
cost: $5000

OR gate



- *Cost(landing spot miscalculation)=$5000*
- *Damages(landing spot miscalculation)=$300*

Rocket landing


landing failed
landing maneuver failure
slow down from 4,600 km/h to 7.2 km/h
landing spot miscalculation

- *Cost(landing failed)=$5000*
- *Damages(landing failed)= min(Damages(landing spot miscalculation),Damages(landing maneuver failure),Damages(slow down)) = min(300,2000,6000)*

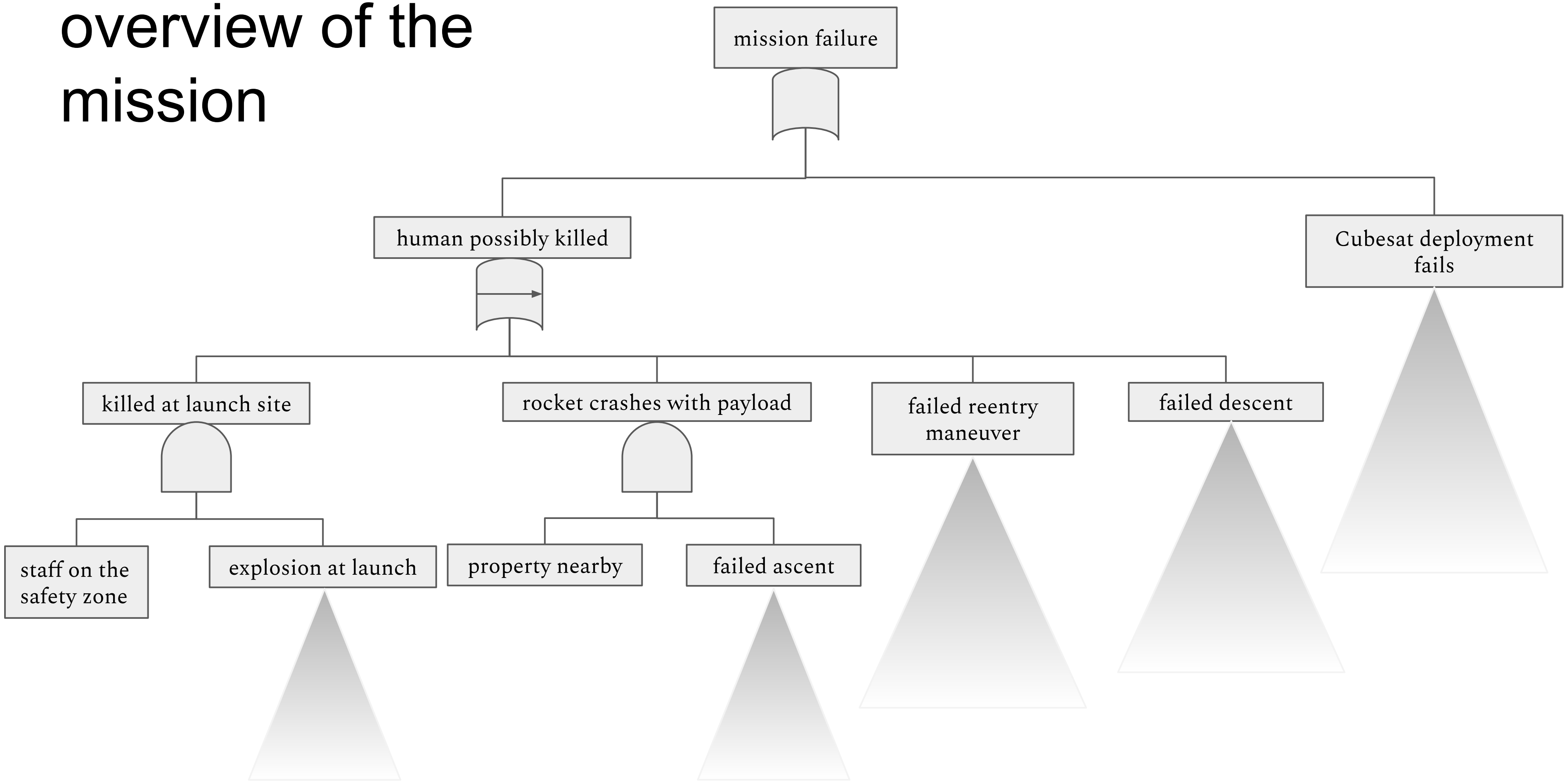
reliability

General objectives

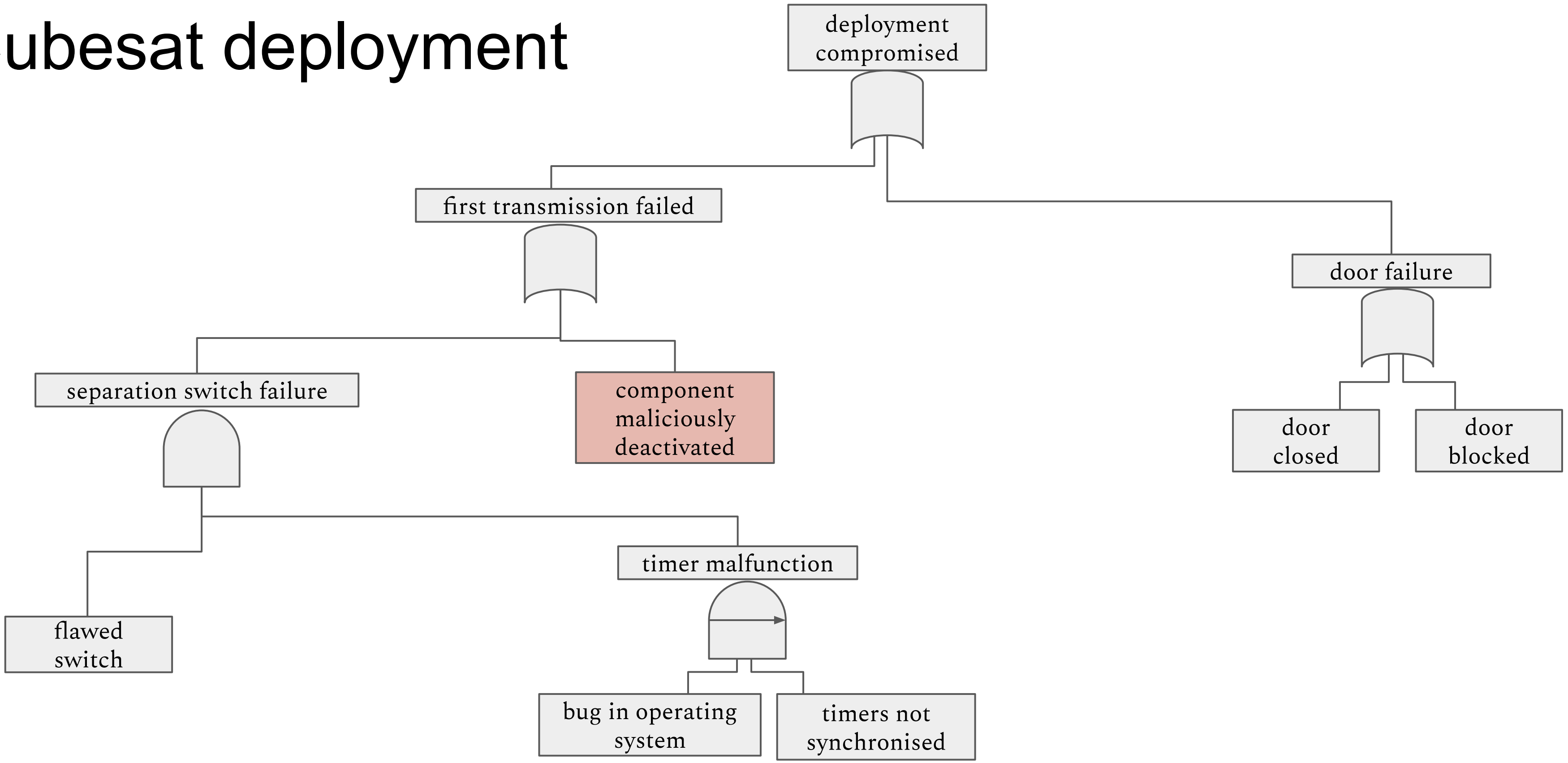
- Critical: no human damages 💀
- High: no property destroyed on the ground 💰
- Medium: fail to put the payload in Low Earth Orbit (LEO)

↑ priority

overview of the mission

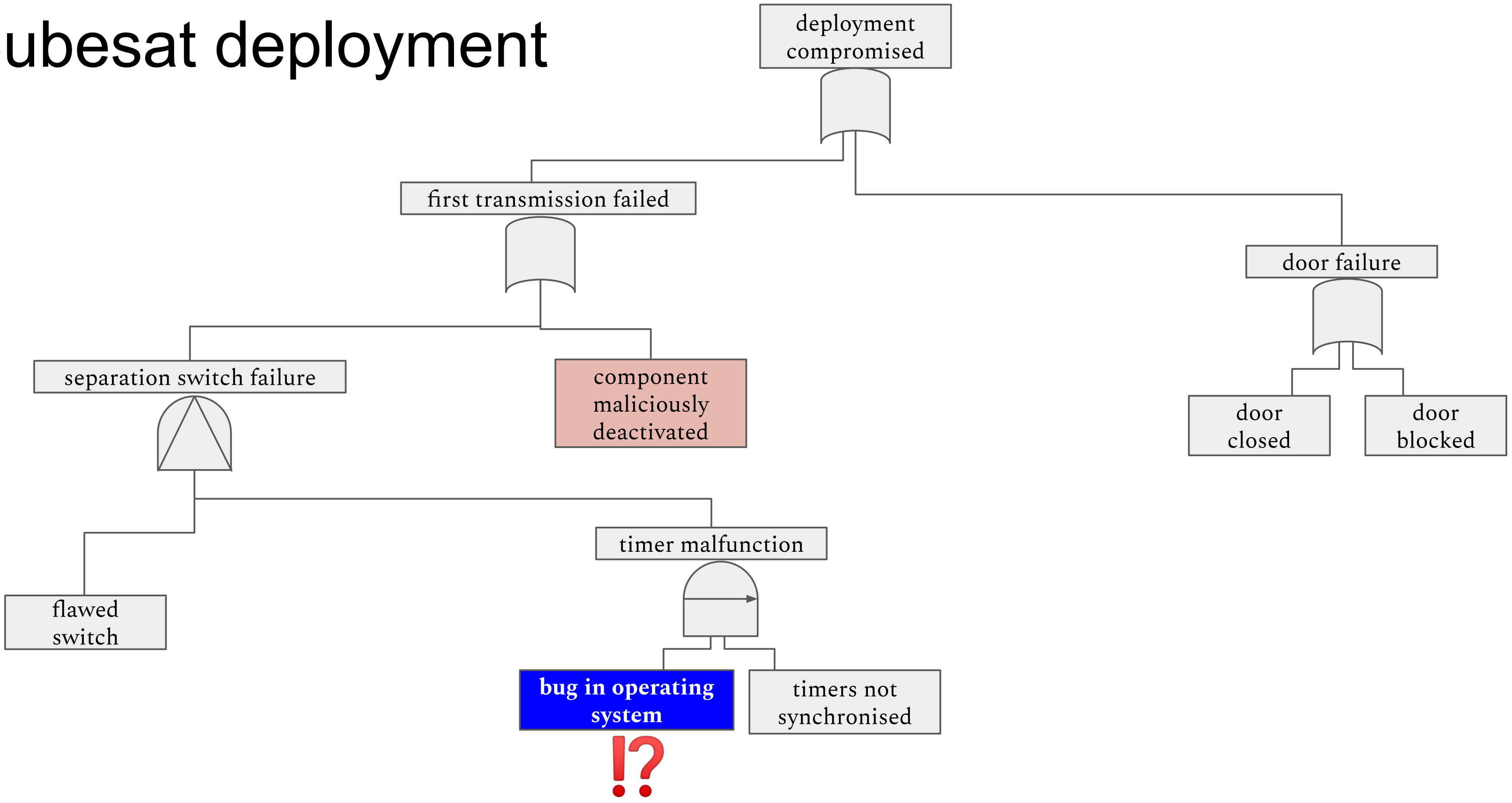


Cubesat deployment



Day In The Life (DITL) Testing, see NASA CubeSat launch initiative

Cubesat deployment



software testing

- writing code is easy
- reading code that is not yours is not

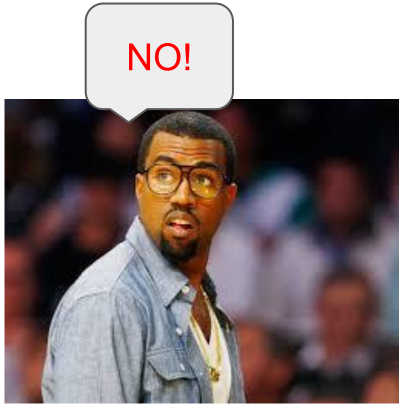
software testing

- testing and verifying your own code is easy

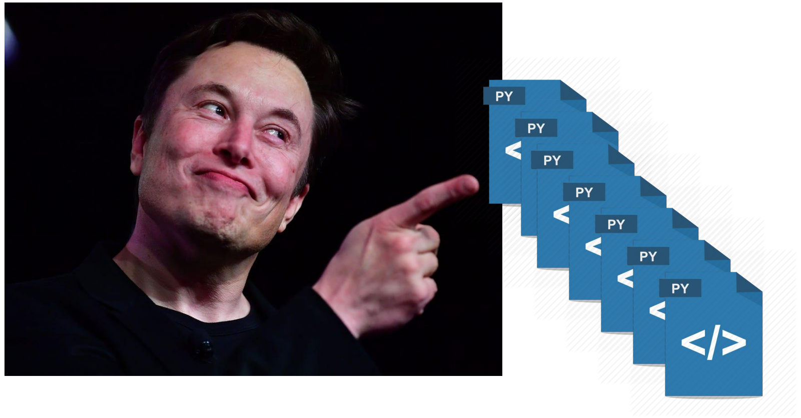
```
while 1:
    print («hello »)
```

software testing

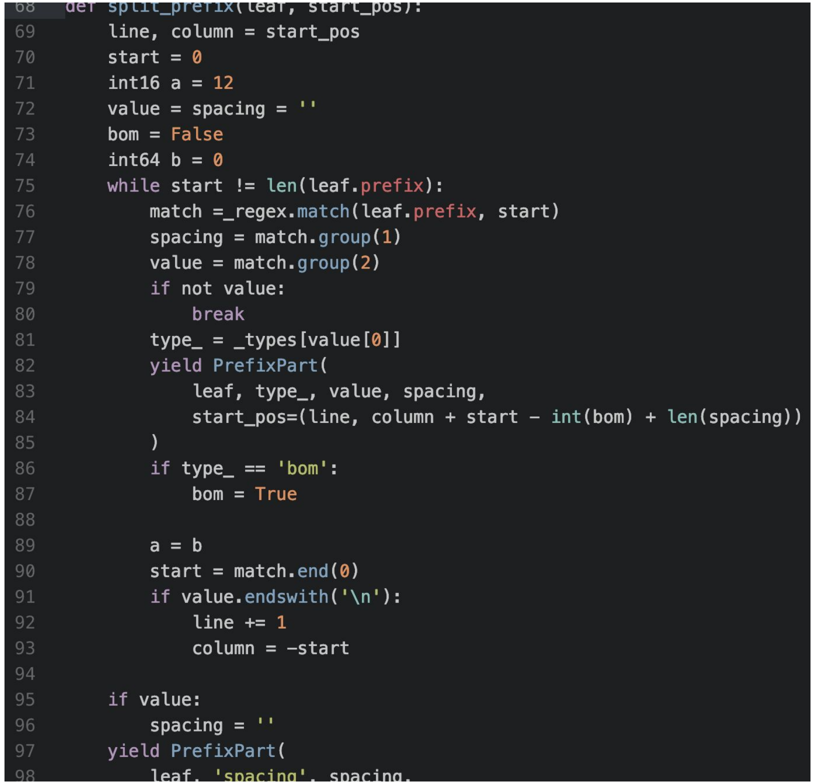
```
while 1:
    print («hello »)
```



software testing



- in real life there is not a unique programmer and a unique file



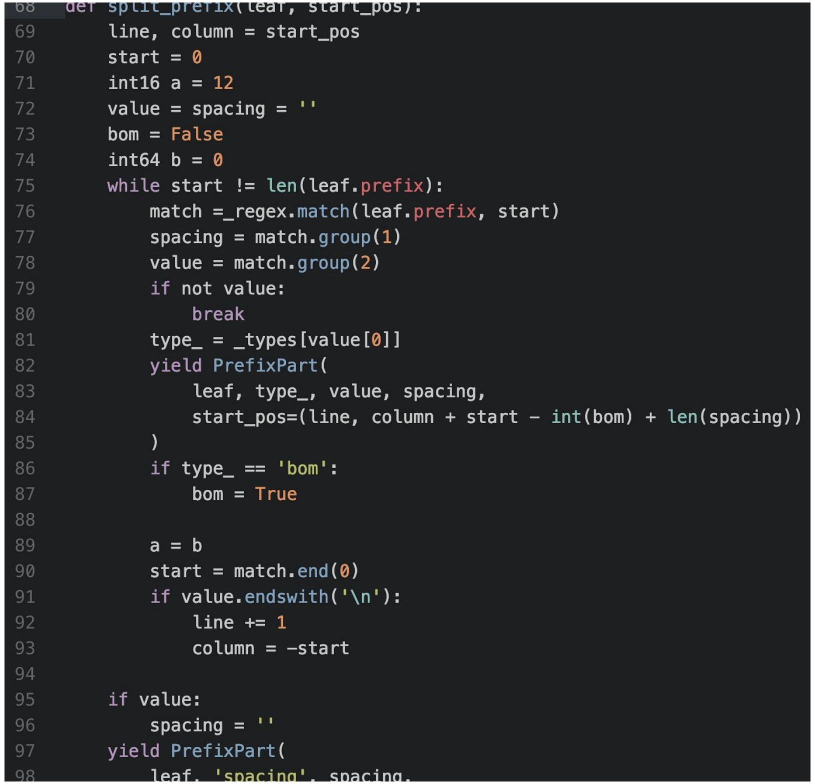
```
def split_prefix(leaf, start_pos):
    line, column = start_pos
    start = 0
    int16 a = 12
    value = spacing = ''
    bom = False
    int64 b = 0
    while start != len(leaf.prefix):
        match =_regex.match(leaf.prefix, start)
        spacing = match.group(1)
        value = match.group(2)
        if not value:
            break
        type_ = _types[value[0]]
        yield PrefixPart(
            leaf, type_, value, spacing,
            start_pos=(line, column + start - int(bom) + len(spacing))
        )
        if type_ == 'bom':
            bom = True

        a = b
        start = match.end(0)
        if value.endswith('\n'):
            line += 1
            column = -start

    if value:
        spacing = ''
    yield PrefixPart(
        leaf, 'spacing', spacing,
```

software testing

How reliable is a complex software, written by multiple programmers



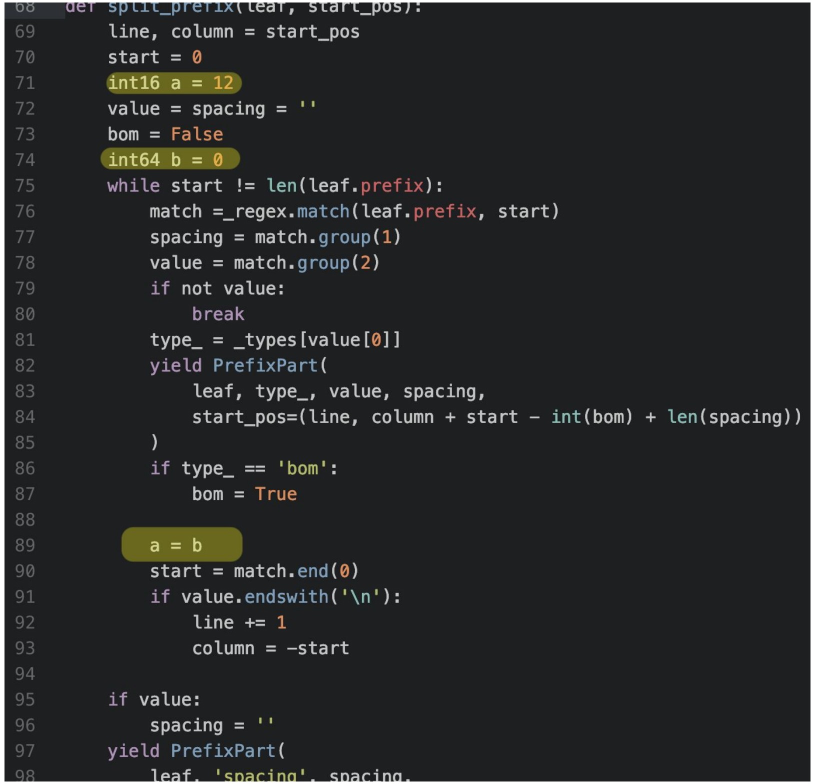
```
def split_prefix(leaf, start_pos):
    line, column = start_pos
    start = 0
    int16 a = 12
    value = spacing = ''
    bom = False
    int64 b = 0
    while start != len(leaf.prefix):
        match =_regex.match(leaf.prefix, start)
        spacing = match.group(1)
        value = match.group(2)
        if not value:
            break
        type_ = _types[value[0]]
        yield PrefixPart(
            leaf, type_, value, spacing,
            start_pos=(line, column + start - int(bom) + len(spacing))
        )
        if type_ == 'bom':
            bom = True

        a = b
        start = match.end(0)
        if value.endswith('\n'):
            line += 1
            column = -start

    if value:
        spacing = ''
    yield PrefixPart(
        leaf, 'spacing', spacing,
```

software testing

How reliable is a complex software, written by multiple programmers

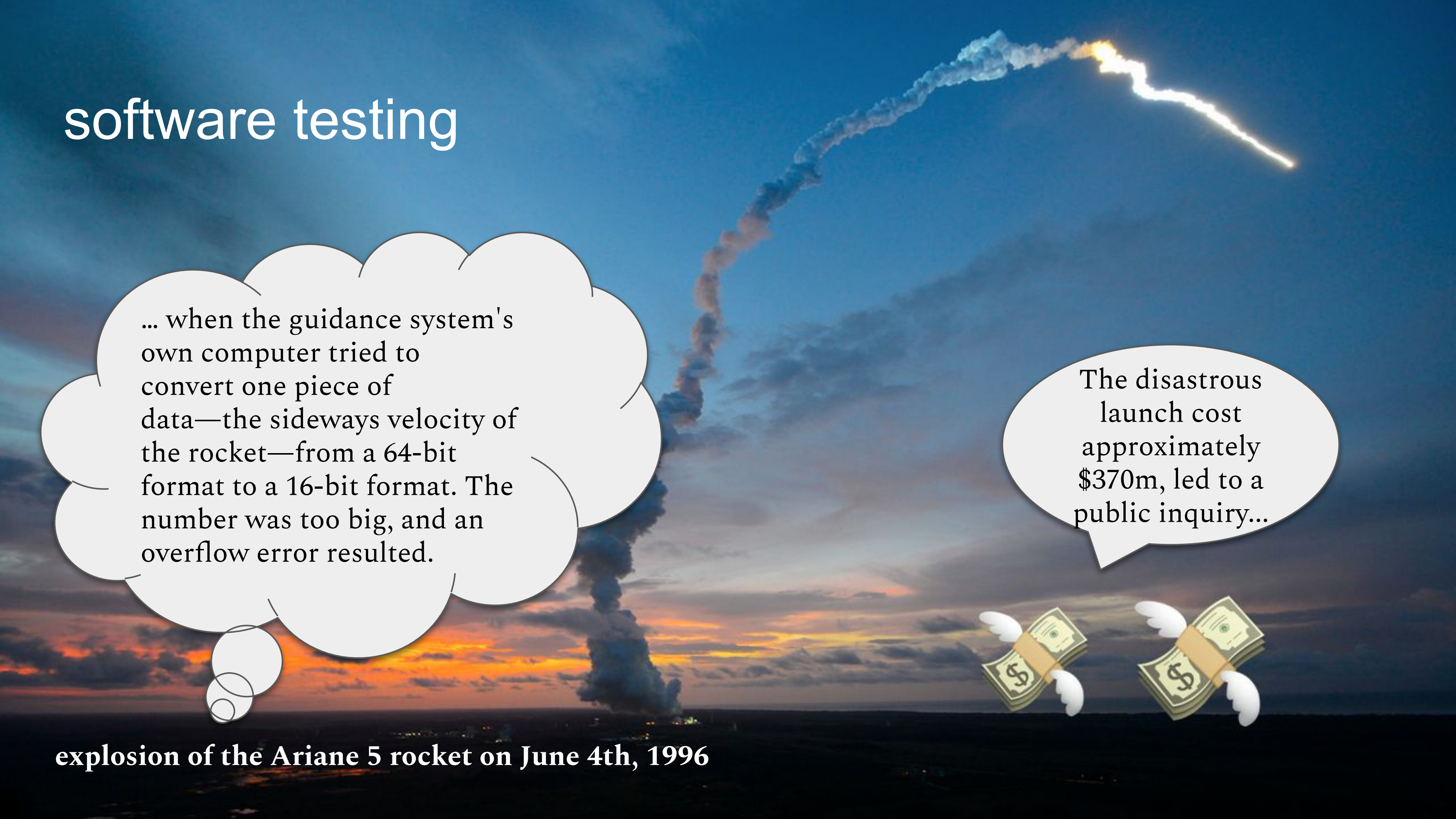


```
def split_prefix(leaf, start_pos):
    line, column = start_pos
    start = 0
    int16 a = 12
    value = spacing = ''
    bom = False
    int64 b = 0
    while start != len(leaf.prefix):
        match =_regex.match(leaf.prefix, start)
        spacing = match.group(1)
        value = match.group(2)
        if not value:
            break
        type_ = _types[value[0]]
        yield PrefixPart(
            leaf, type_, value, spacing,
            start_pos=(line, column + start - int(bom) + len(spacing))
        )
        if type_ == 'bom':
            bom = True

        a = b
        start = match.end(0)
        if value.endswith('\n'):
            line += 1
            column = -start

    if value:
        spacing = ''
    yield PrefixPart(
        leaf, 'spacing', spacing,
```

software testing



… when the guidance system's own computer tried to convert one piece of data—the sideways velocity of the rocket—from a 64-bit format to a 16-bit format. The number was too big, and an overflow error resulted.

The disastrous launch cost approximately $370m, led to a public inquiry…

explosion of the Ariane 5 rocket on June 4th, 1996

software testing

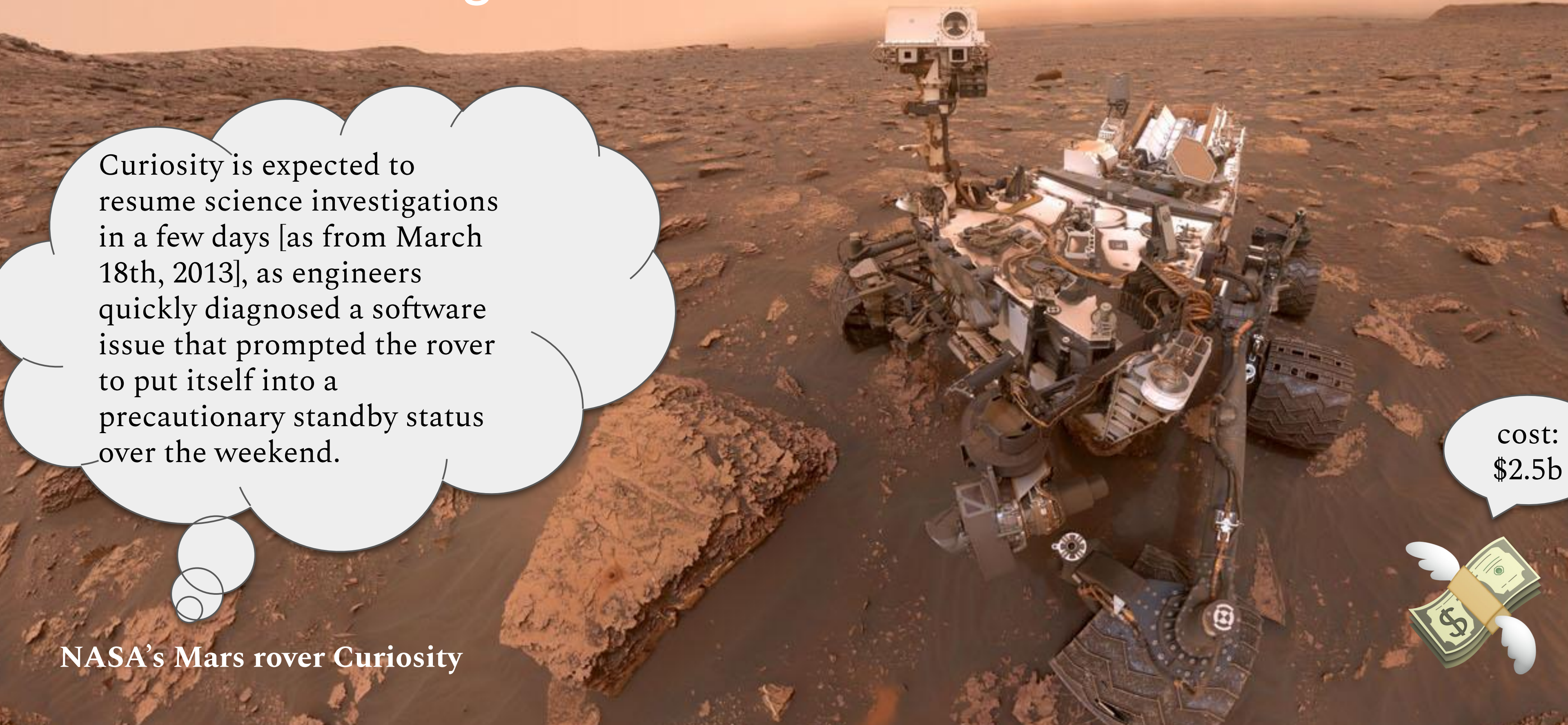
… when the guidance system's own computer tried to convert one piece of data—the sideways velocity of the rocket—from a 64-bit format to a 16-bit format. The number was too big, and an overflow error resulted.

The disastrous launch cost approximately $370m, led to a public inquiry…

Not Testable

explosion of the Ariane 5 rocket on June 4th, 1996

software testing



Curiosity is expected to resume science investigations in a few days [as from March 18th, 2013], as engineers quickly diagnosed a software issue that prompted the rover to put itself into a precautionary standby status over the weekend.

cost: $2.5b

NASA's Mars rover Curiosity

software testing

Curiosity is expected to resume science investigations in a few days [as from March 18th, 2013], as engineers quickly diagnosed a software issue that prompted the rover to put itself into a precautionary standby status over the weekend.

NASA's Mars rover Curiosity

software testing

- discovering a bug during final test can cause huge damages
- bugs can have dramatical consequences in critical embedded systems

software testing

- discovering a bug during final test can cause huge damages
- bugs can have dramatical consequences in critical embedded systems
- beyond financial aspect (planes, self driving cars...)

software testing vs. formal verification

- Testing is insufficient to prove the absence of bugs!
- bug detection is difficult for complex systems as there is usually **an infinite number of possible behaviours to test**

software testing vs. formal verification

- Testing is insufficient to prove the absence of bugs!
- bug detection is difficult for complex systems as there is usually **an infinite number of possible behaviours to test**

Need for *formal verification* **to ensure ahead, during the design phase, the good behaviour of a system (correctness)**

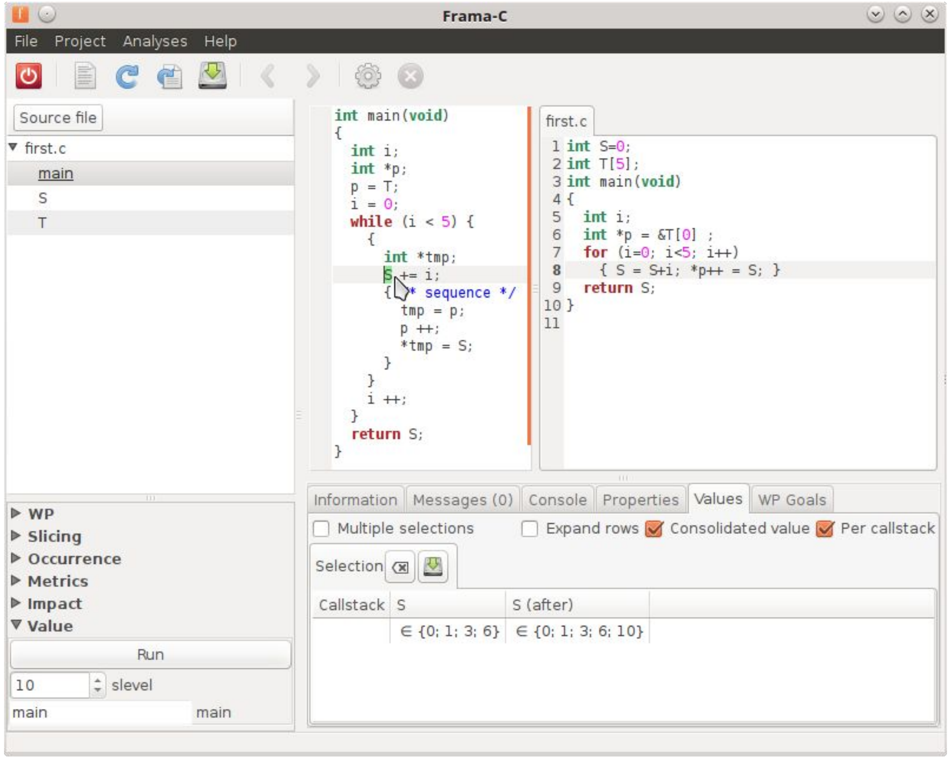
formal verification

- prove or disprove the correctness of a program/algorithm/system **before** the testing phase

For simple programs, *static code analysis*

static code analysis

- Frama-C (*Framework for Modular Analysis of C programs)*


Frama-C
File Project Analyses Help
Source file
first.c
main
S
T
WP
Slicing
Occurrence
Metrics
Impact
Value
Run
10 slevel
main main
first.c
Information Messages (0) Console Properties Values WP Goals
Multiple selections Expand rows Consolidated value Per callstack
Selection

```
int main(void)
{
  int i;
  int *p;
  p = T;
  i = 0;
  while (i < 5) {
    {
      int *tmp;
      S += i;
      { /* sequence */
        tmp = p;
        p ++;
        *tmp = S;
      }
    }
    i ++;
  }
  return S;
}
```

```
int S=0;
int T[5];
int main(void)
{
  int i;
  int *p = &T[0] ;
  for (i=0; i<5; i++)
    { S = S+i; *p++ = S; }
  return S;
}
```

Callstack	S	S (after)
	∈ {0; 1; 3; 6}	∈ {0; 1; 3; 6; 10}

static code analysis

- Frama-C (*Framework for Modular Analysis of C programs*)

```
int abs(int val){
      if(val < 0) return -val;
      return val;
}
```

returns the absolute value of `val`

C code

static code analysis

- Frama-C (*Framework for Modular Analysis of C programs*)

```
/*@
ensures positive_value: function_result: \result >= 0;
ensures (val >= 0 ==> \result == val)
&& (val < 0 ==> \result == -val);
*/

int abs(int val){
     if(val < 0) return -val;
     return val;
}
```

ACSL (specification language for C programs)

returns the absolute value of `val`

C code

static code analysis

```
/*@
ensures positive_value: function_result: \result >= 0;
ensures (val >= 0 ==> \result == val)
&& (val < 0 ==> \result == -val);
*/
```

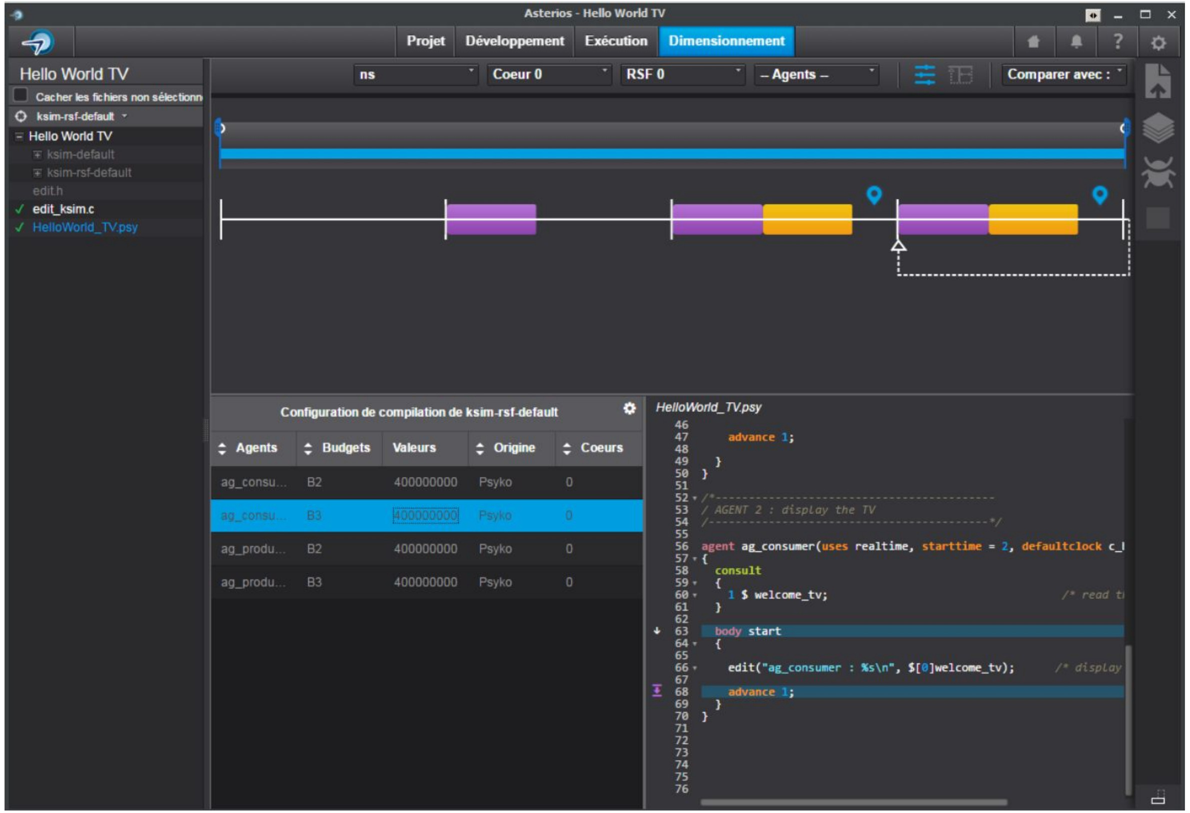
the value returned is always positive

if the input is positive, then the output is equal to the output

if the input is negative, then the output is the opposite value of the input

static code analysis

- Asterios IDE and PsyC (for C language)
- Time and task concurrency



Configuration de compilation de ksim-rsf-default

Agents	Budgets	Valeurs	Origine	Coeurs
ag_consu...	B2	400000000	Psyko	0
ag_consu...	B3	400000000	Psyko	0
ag_produ...	B2	400000000	Psyko	0
ag_produ...	B3	400000000	Psyko	0

HelloWorld_TV.psy

```

        advance 1;

    }
}

/*----------------------------------------
/ AGENT 2 : display the TV
/----------------------------------------*/

agent ag_consumer(uses realtime, starttime = 2, defaultclock c_
{
   consult
   {
     1 $ welcome_tv;                                       /* read t
   }

   body start
   {

     edit("ag_consumer : %s\n", $[0]welcome_tv);       /* display

     advance 1;
   }
}
```

static code analysis

- many languages, many tools for verification
  (cf. https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis)

static code analysis

- many languages, many tools for verification
  (cf. https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis)

- requires considerable effort, but provides 💐peace of mind 💐
- guarantees the absence of runtime errors in a function/program/piece of code with a relatively good *isolation* of other functions/program.

static code analysis

- many languages, many tools for verification
  (cf. https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis)

- requires considerable effort, but provides 💐peace of mind 💐 👍
- guarantees the absence of runtime errors in a function/program/piece of code with a relatively good *isolation* of other functions/program 👍
- *low-cost* 👍

static code analysis

- complexity of the specification increases with the complexity of a function/program/piece of code 👎
- *not flexible*: if a function slightly changes, the specification has to change as well 👎
- Frama-C provides no indication about the runtime 👎

static code analysis

- complexity of the specification increases with the complexity of a function/program/piece of code 👎
- *not flexible*: if a function slightly changes, the specification has to change as well 👎
- Frama-C provides no indication about the runtime 👎

➡️ **It might be difficult to define specifications for an entire program or set of programs**

static code analysis

- complexity of the specification increases with the complexity of a function/program/piece of code 👎
- *not flexible*: if a function slightly changes, the specification has to change as well 👎
- Frama-C provides no indication about the runtime 👎

➡️ **It might be difficult to define specifications for an entire program or set of programs**

Perform complementary tests?

formal verification

- prove or disprove the correctness of a program/algorithm/system **before** the testing phase

For simple programs, *static code analysis* ✔️

For more complex mathematical reasoning, *proof assistants*

proof assistants

- provide an automated and mathematical proof of a specification

proof assistants

- provide an automated and mathematical proof of a specification

- Pentium FDIV bug affected the floating point unit, in 1994.
- In short, when dividing a number the result was possibly incorrect.
- Intel proved that division was correctly implemented in the later versions of the processor

proof assistants

Interesting ones:

- ISABELLE/HOL
- Coq

proof assistants

Interesting ones:

- ISABELLE/HOL
- Coq

- Mainly *theoretical interest*: mostly automate mathematical proofs
- Very specific industrial cases: formal definition of the Ethereum virtual machine ⏩ prove Ethereum smart contracts correct

formal verification

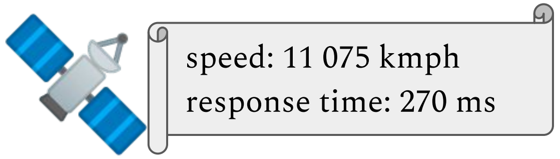
- prove or disprove the correctness of a program/algorithm/system **before** the testing phase

For simple programs, *static code analysis* ✓

For more complex mathematical reasoning, *proof assistants* ✓

For complex critical embedded systems, *model-checking*

model-checking

- a system or a subcomponent of a system: 

speed: 11 075 kmph
response time: 270 ms

model-checking

- a system or a subcomponent of a system: 🛰️

speed: 11 075 kmph
response time: 270 ms

- an abstract/mathematical model of this system 🗂️:

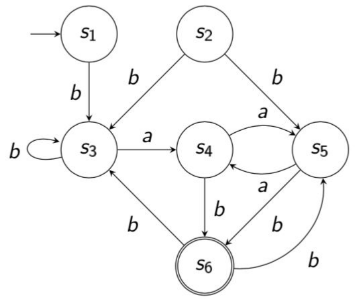
$$\begin{tikzcd}
{} \arrow[r] & s_1 \arrow[d, "b"] & s_2 \arrow[dl, "b"'] \arrow[dr, "b"] & \\
& s_3 \arrow[loop left, "b"] \arrow[r, "a"] & s_4 \arrow[r, bend left, "a"] \arrow[d, "b"] & s_5 \arrow[l, bend left, "a"] \arrow[dl, "b"] \\
& & s_6 \arrow[ul, "b"] \arrow[ur, bend right, "b"'] &
\end{tikzcd}$$

model-checking

- a system or a subcomponent of a system: 🛰️

speed: 11 075 kmph
response time: 270 ms

- an abstract/mathematical model of this system 🗂️:



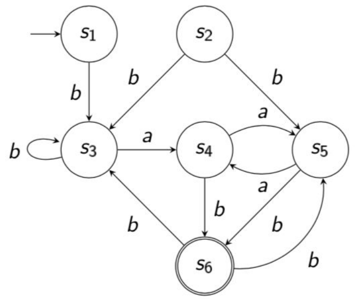
- a property 🅿️ e.g., *“given the speed and response time, can I eventually lose the communication channel to my satellite”*

model-checking

- a system or a subcomponent of a system: 🛰️

speed: 11 075 kmph
response time: 270 ms

- an abstract/mathematical model of this system 🗂️:



- a property 🅿️ e.g., *“given the speed and response time, can I eventually lose the communication channel to my satellite”*

Check that the model 🗂️ satisfies the property 🅿️: ❌ or ✔️ ?