The Dangers of Inconsistency

Selmer Bringsjord

Mike Giancola

RAIR

Rensselaer AI and Reasoning Lab

Are Humans Rational?
October 21, 2019

Sponsored by

Office of Naval Research ONR Science & Technology

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Rensselaer

Motivation

Motivation

- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies

Motivation

- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies
  - Underlying logic could be inconsistent

Motivation

- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies
  - Underlying logic could be inconsistent
  - Inconsistency could be inherent to the domain

Motivation

- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies
  - Underlying logic could be inconsistent
  - Inconsistency could be inherent to the domain
    - e.g. Belief Revision

Motivation

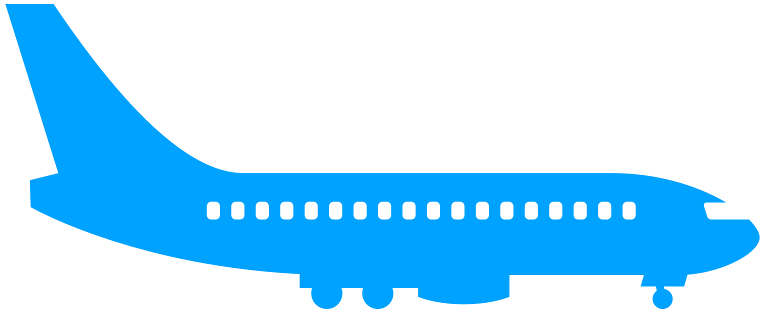
- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies
    - Underlying logic could be inconsistent **Hard!**
    - Inconsistency could be inherent to the domain
        - e.g. Belief Revision

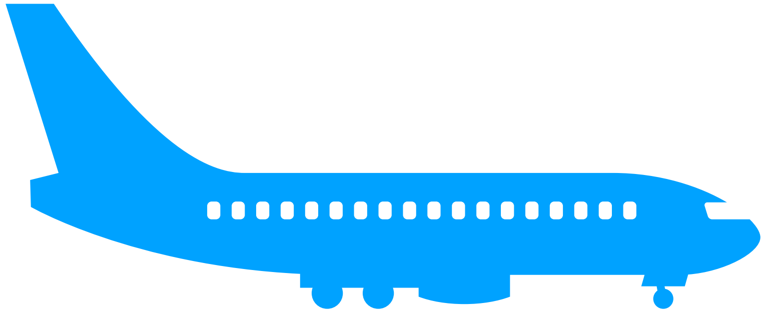
Motivation

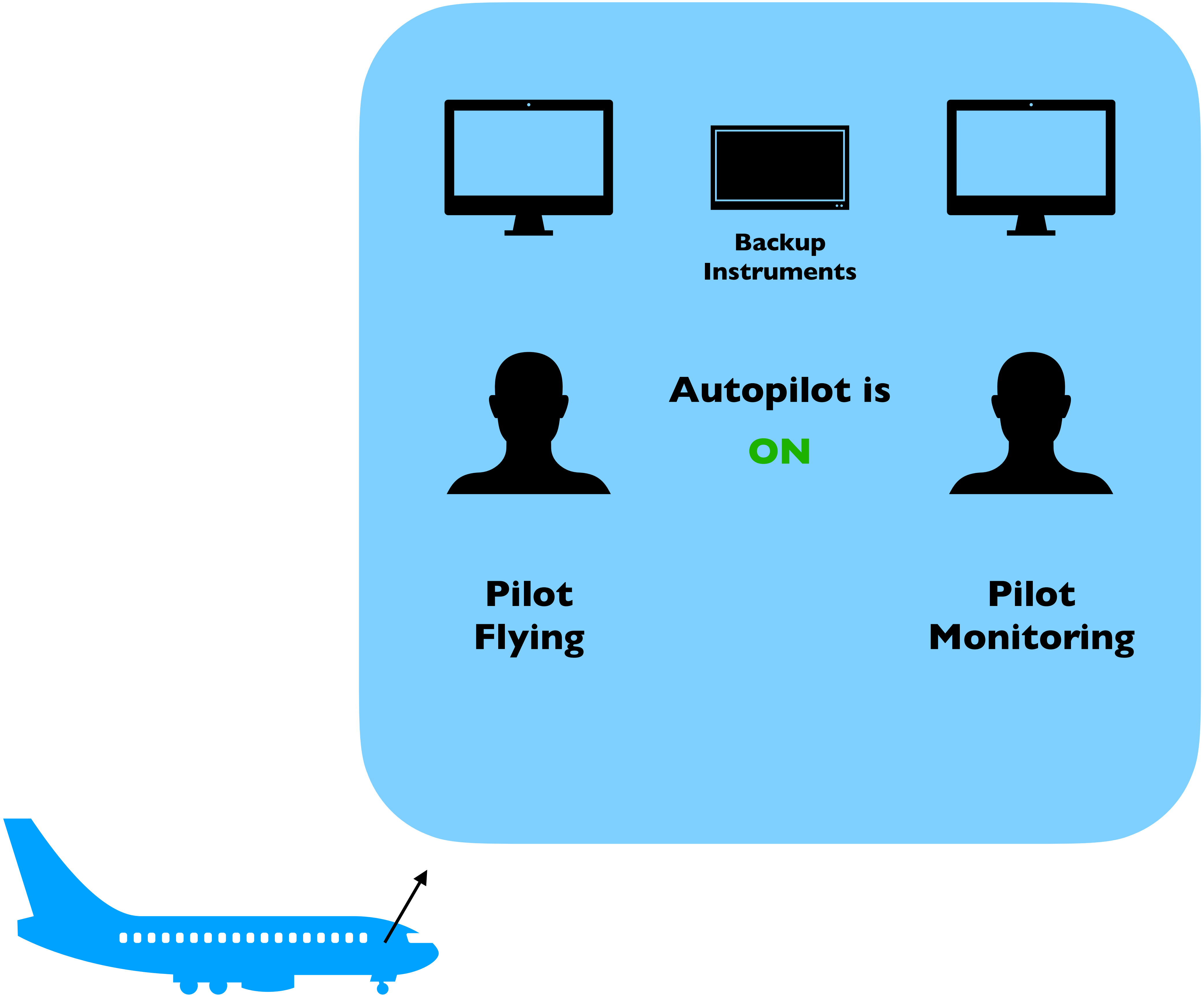
- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies
  - Underlying logic could be inconsistent Hard!
  - Inconsistency could be inherent to the domain
    - e.g. Belief Revision

Motivation

- AI agents utilizing automated reasoning will undoubtedly encounter inconsistencies
    - Underlying logic could be inconsistent Hard!
    - Inconsistency could be inherent to the domain
        - e.g. Belief Revision
- **Goal:** Build a system which can detect inconsistencies and construct solutions





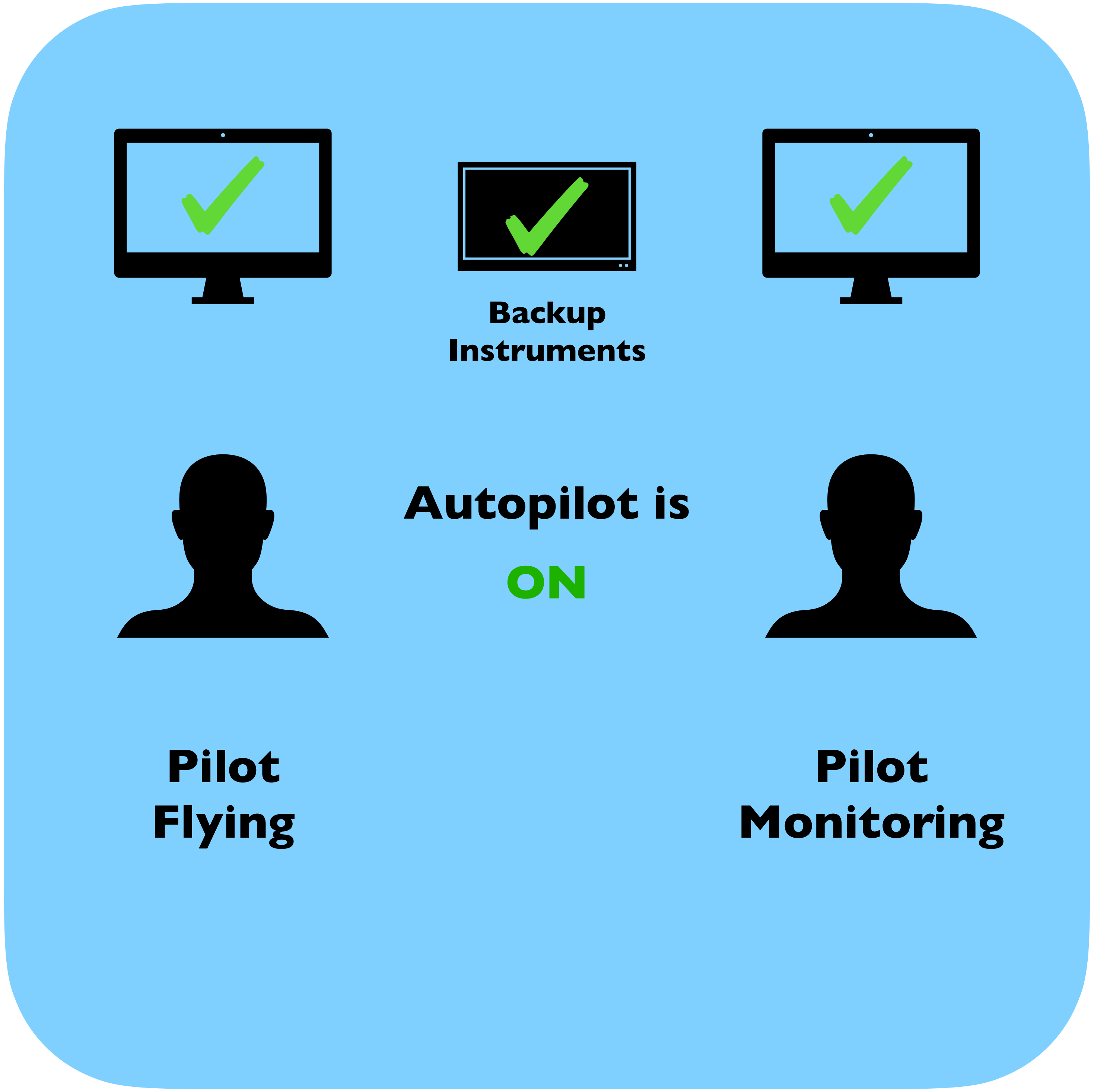


Backup Instruments

Autopilot is ON

Pilot Flying

Pilot Monitoring



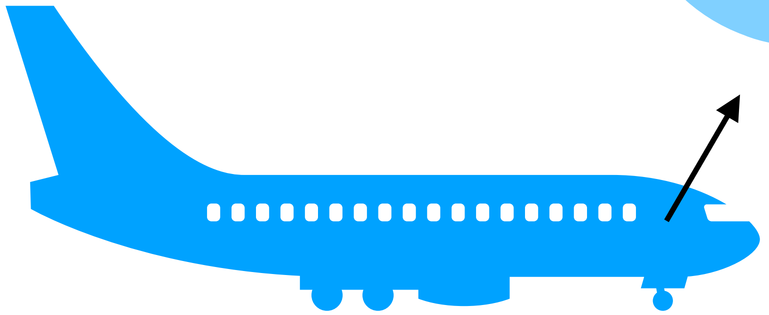
Backup Instruments

Autopilot is

ON

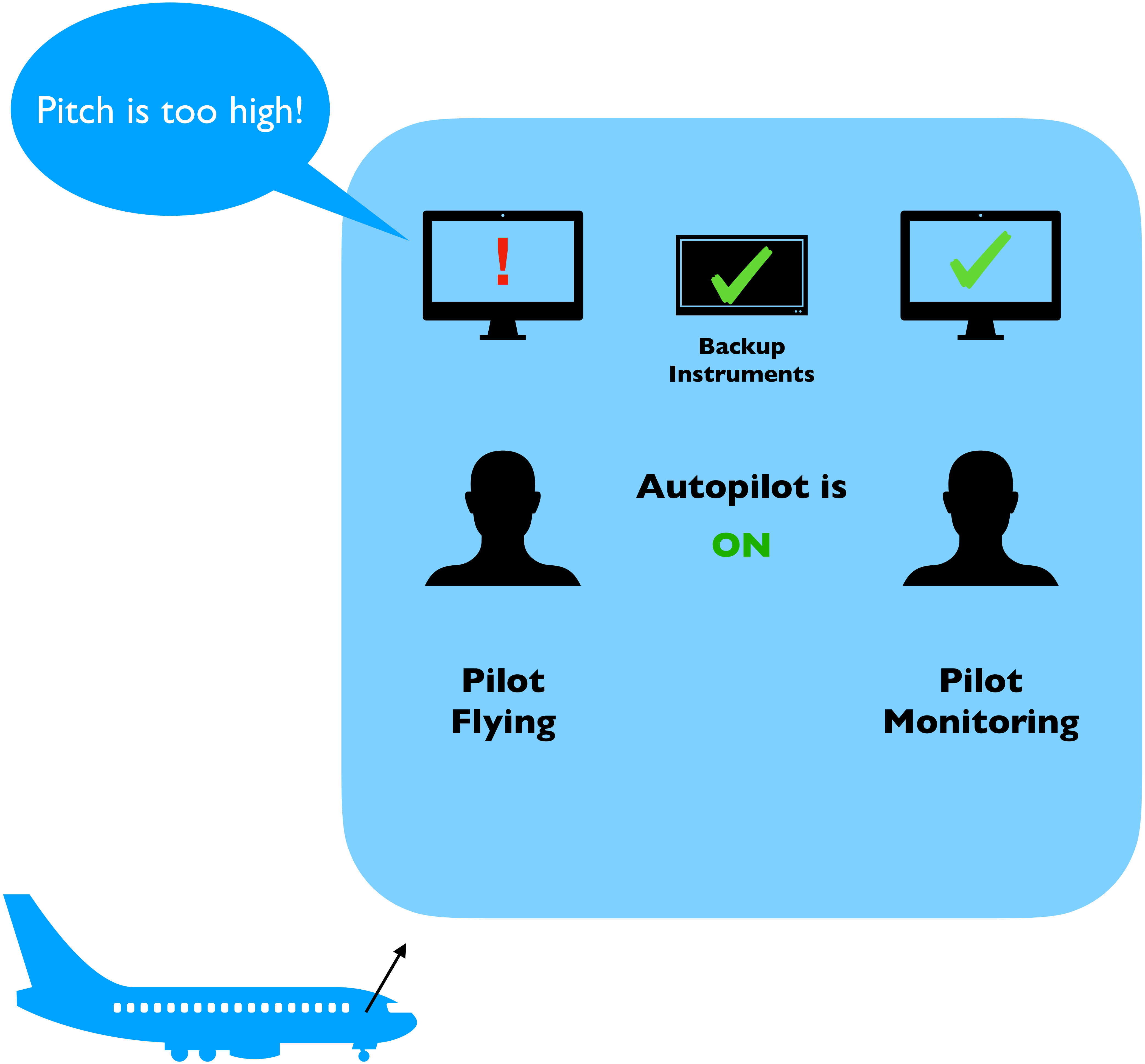
Pilot Flying

Pilot Monitoring



Time:

t_0



Pitch is too high!

Backup Instruments

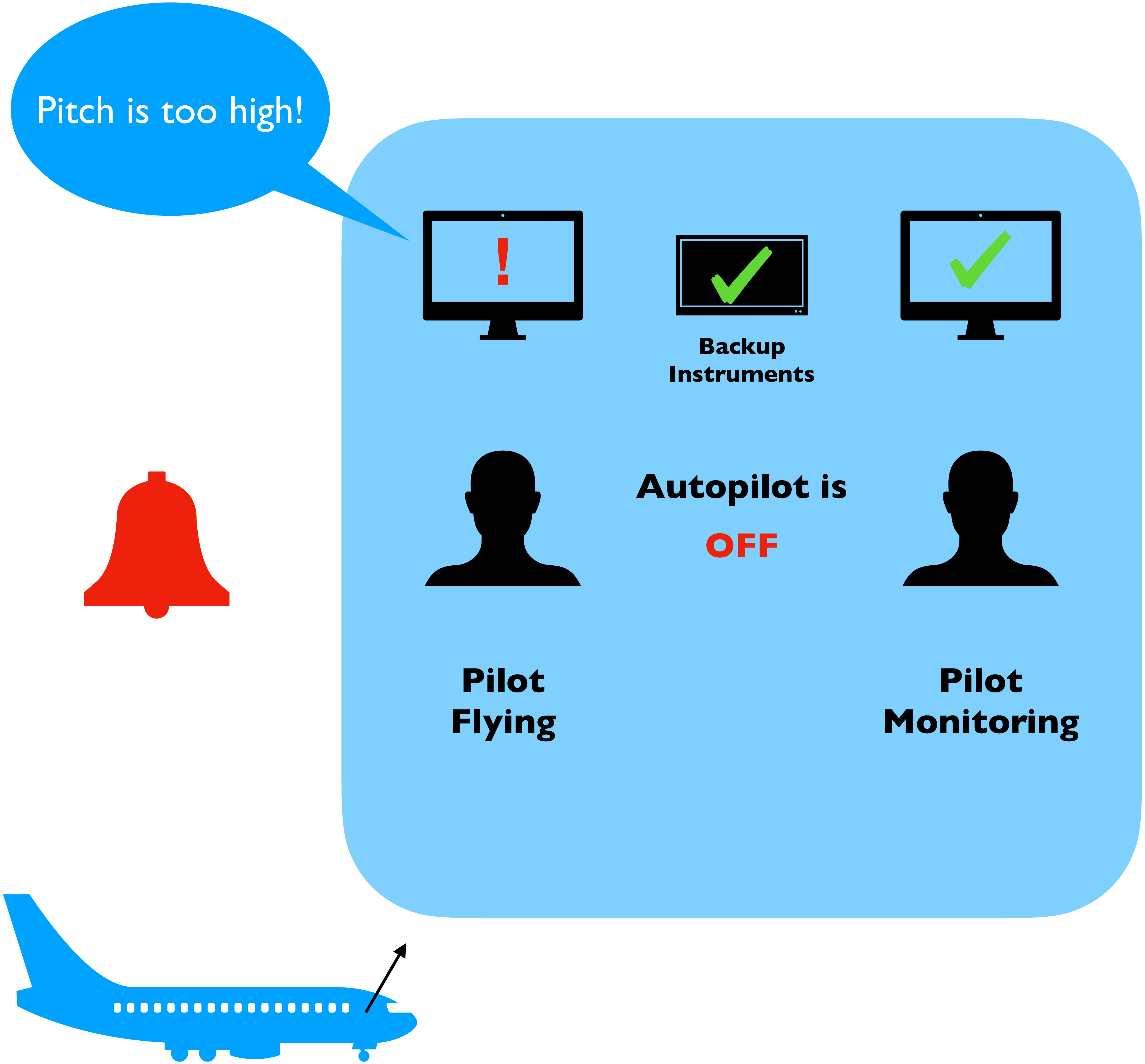
Autopilot is ON

Pilot Flying

Pilot Monitoring

Time:

t_1



Pitch is too high!

Backup Instruments

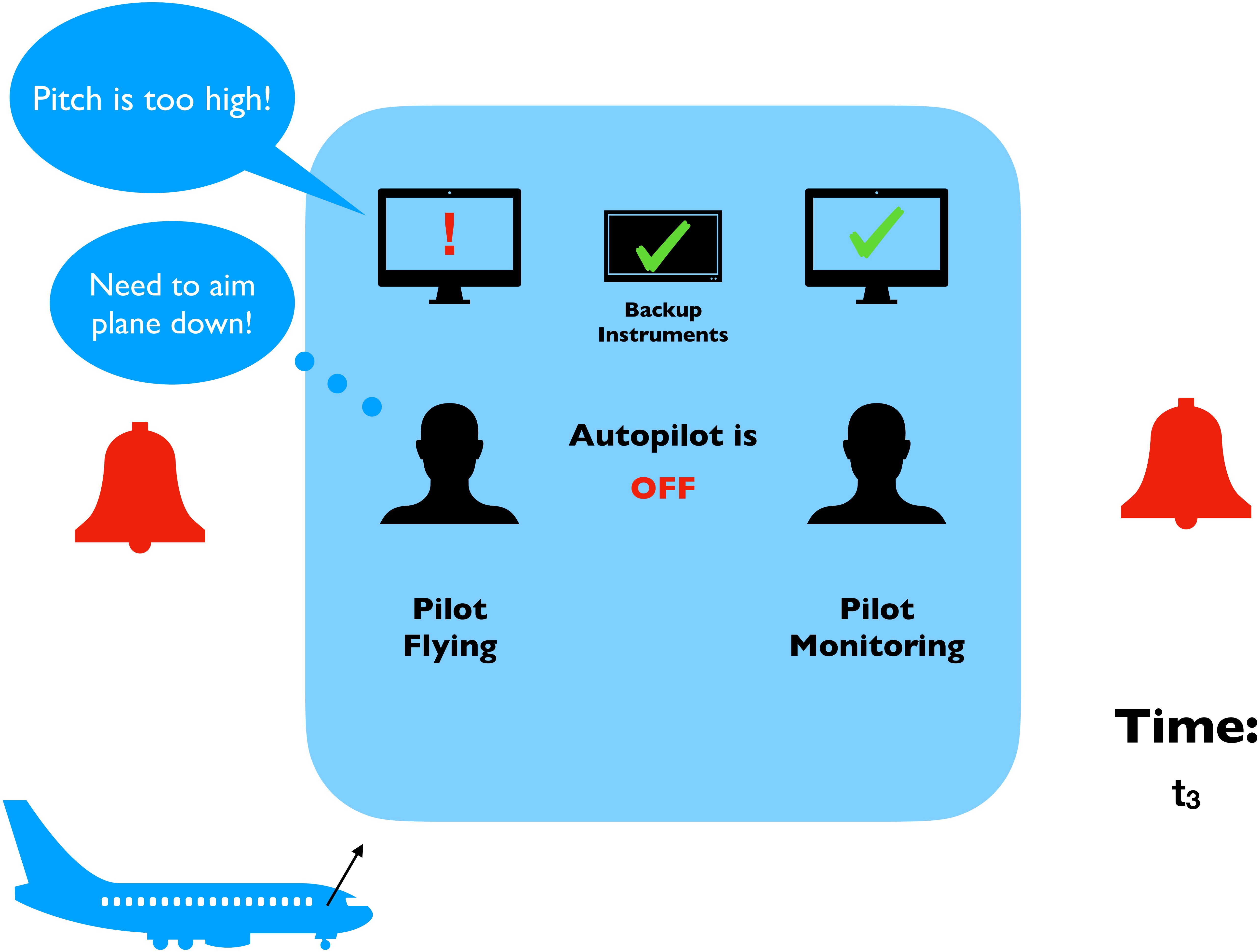
Autopilot is OFF

Pilot Flying

Pilot Monitoring

Time:

t_2



Pitch is too high!

Need to aim plane down!

Backup Instruments

Autopilot is

OFF

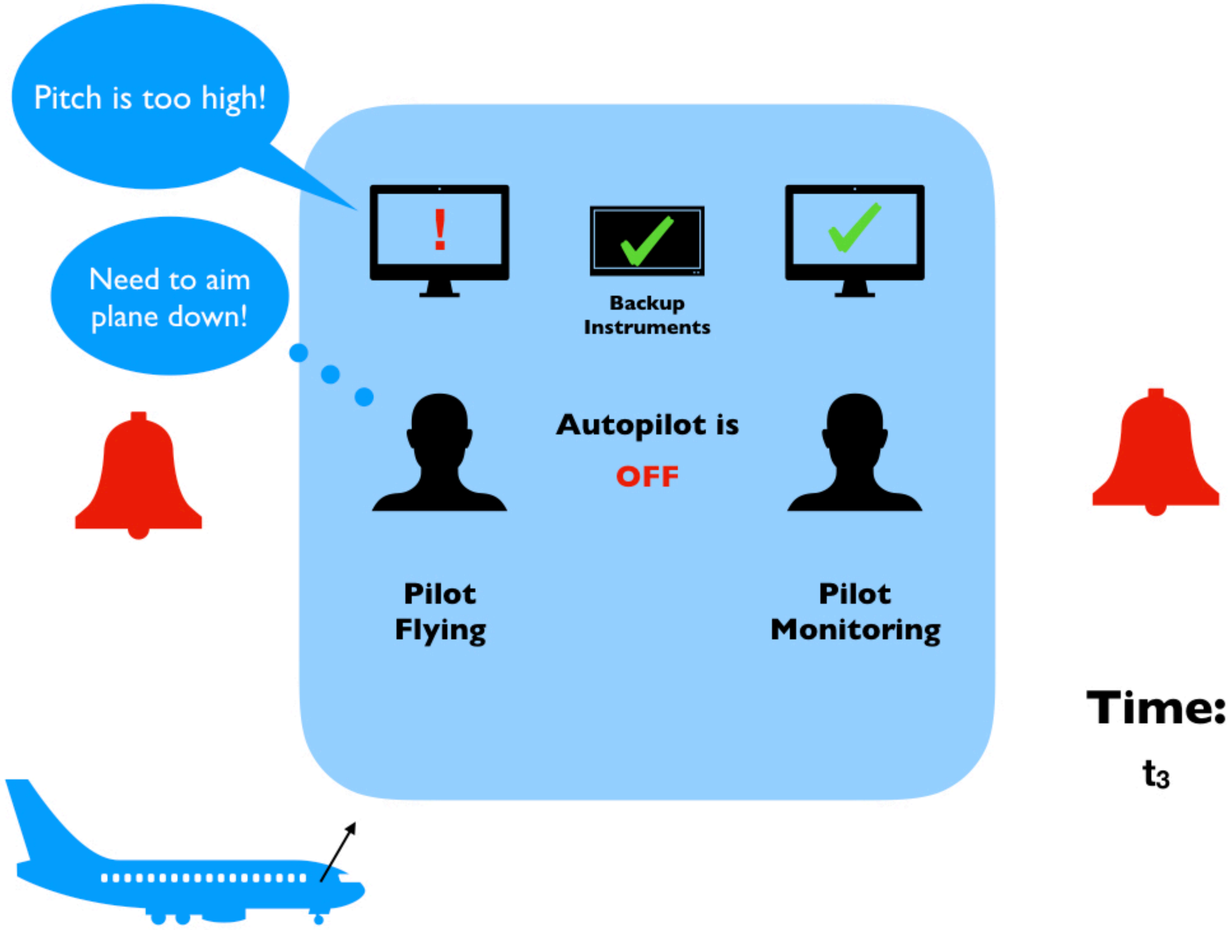
Pilot Flying

Pilot Monitoring

Time:

t_3

What Happened?



Pitch is too high!

Need to aim plane down!

Backup Instruments

Autopilot is **OFF**

Pilot Flying

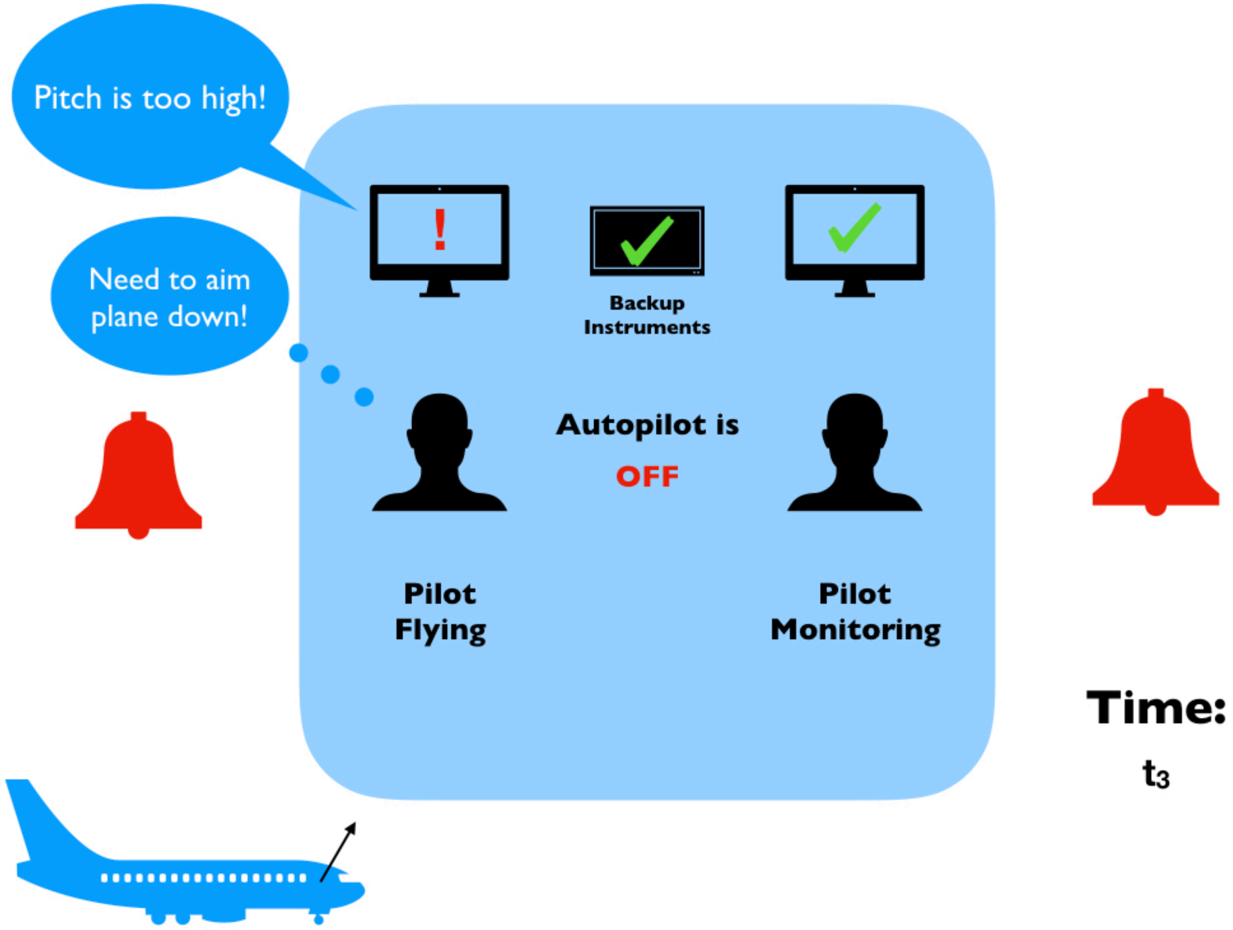
Pilot Monitoring

Time:

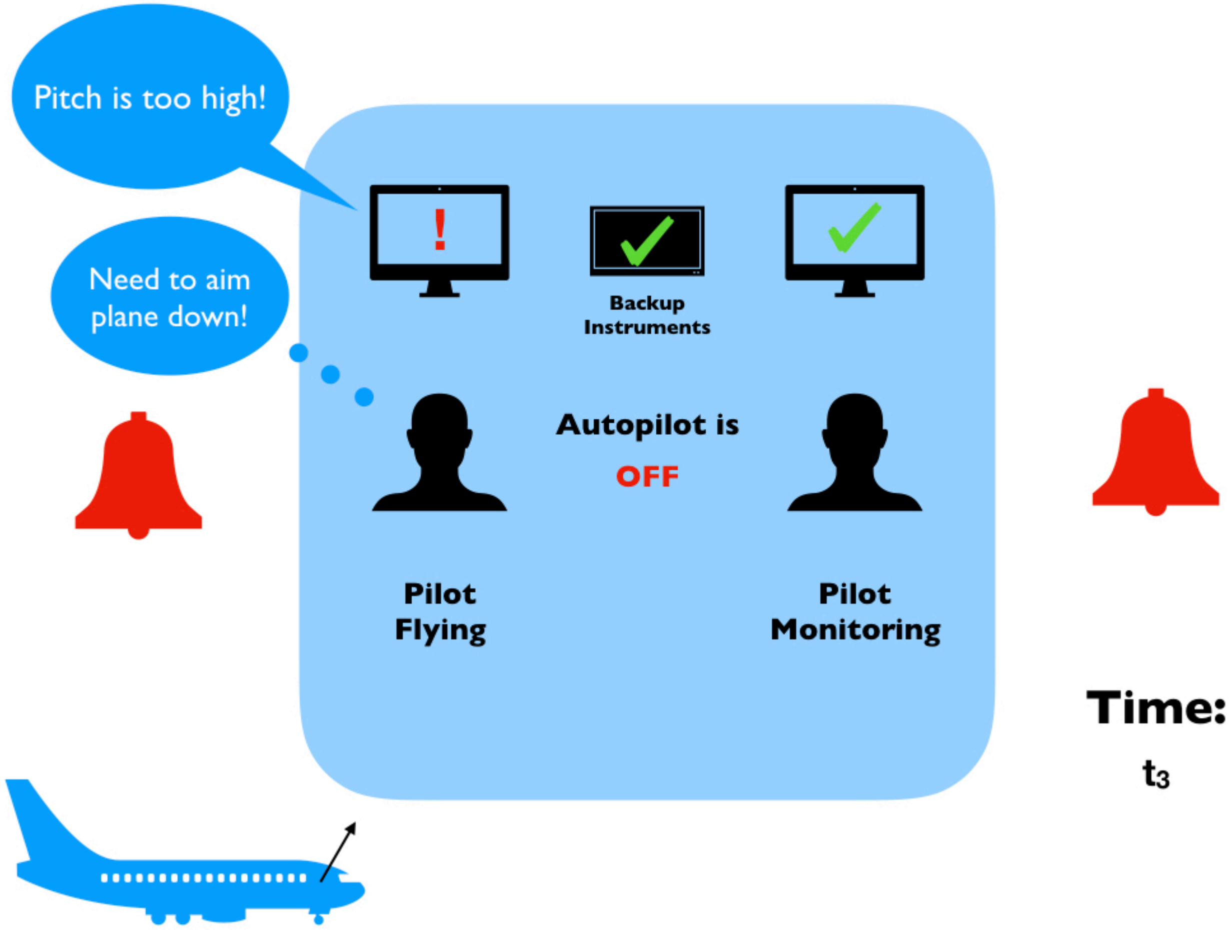
t_3

What Happened?

- Each pilot's display has its own set of sensors

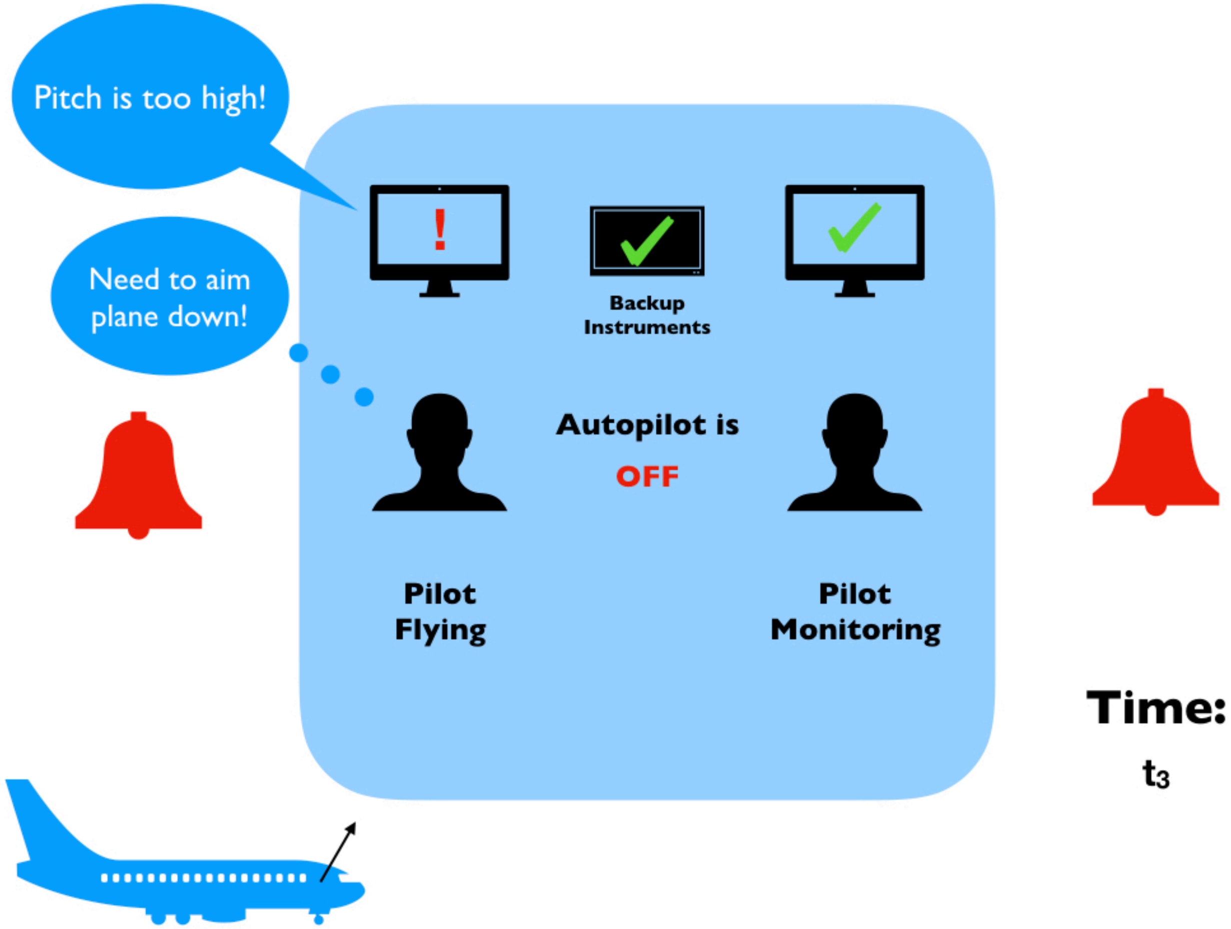


What Happened?



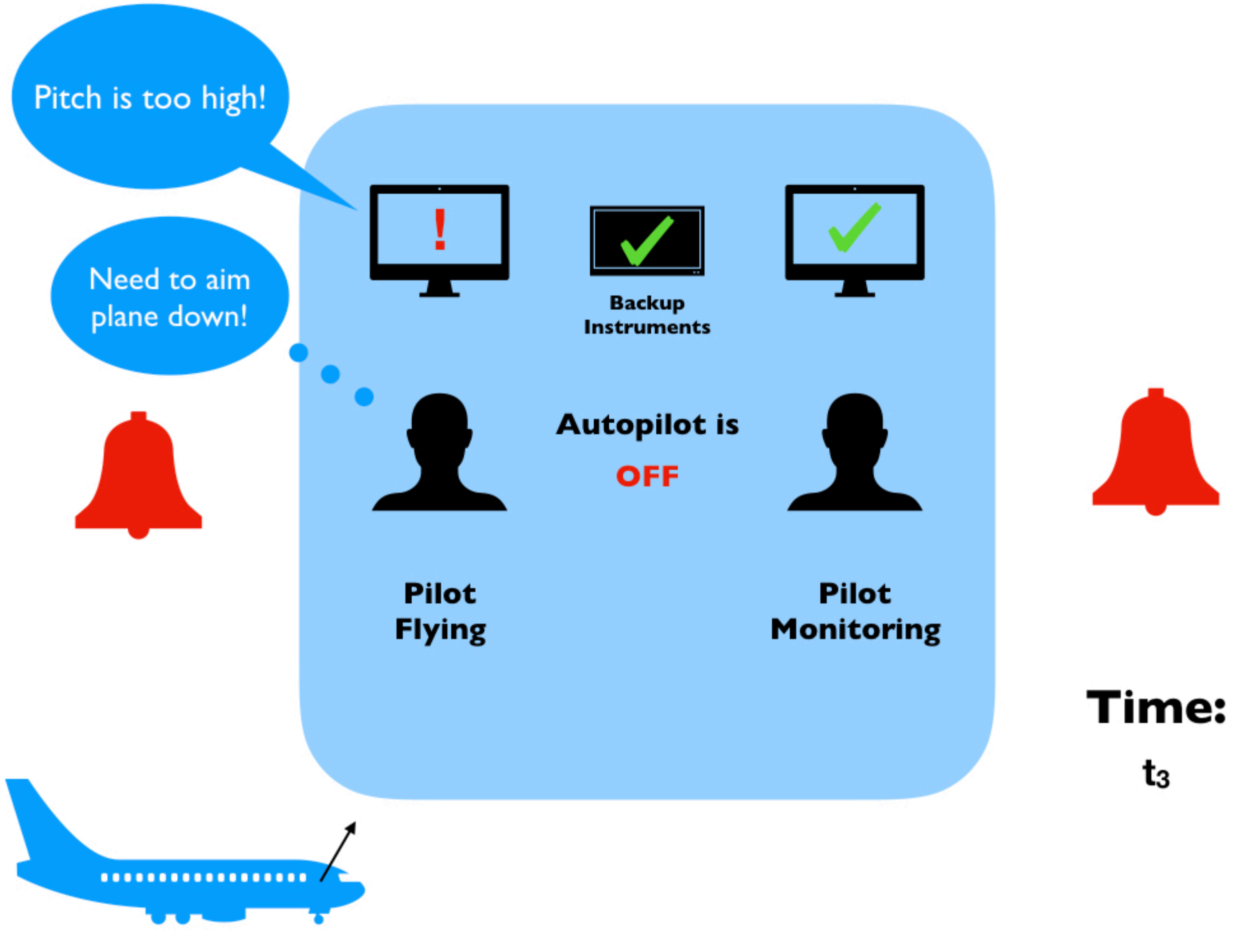
- Each pilot’s display has its own set of sensors
  - A faulty sensor feeding the PF’s display gave an incorrect reading

What Happened?



- Each pilot’s display has its own set of sensors
  - A faulty sensor feeding the PF’s display gave an incorrect reading
  - Typically, a Comparator Function continuously monitors sensor readings

What Happened?



- Each pilot's display has its own set of sensors
  - A faulty sensor feeding the PF's display gave an incorrect reading
  - Typically, a Comparator Function continuously monitors sensor readings
  - This was disabled by a Declutter Function

How could an automated reasoner have helped?

How could an automated reasoner have helped?

- Instantly detect the inconsistency

How could an automated reasoner have helped?

- Instantly detect the inconsistency
    - In this case, notice that Pilot 1's sensor reading seems unusual, and that Pilot 2's reading matches the backup instruments.

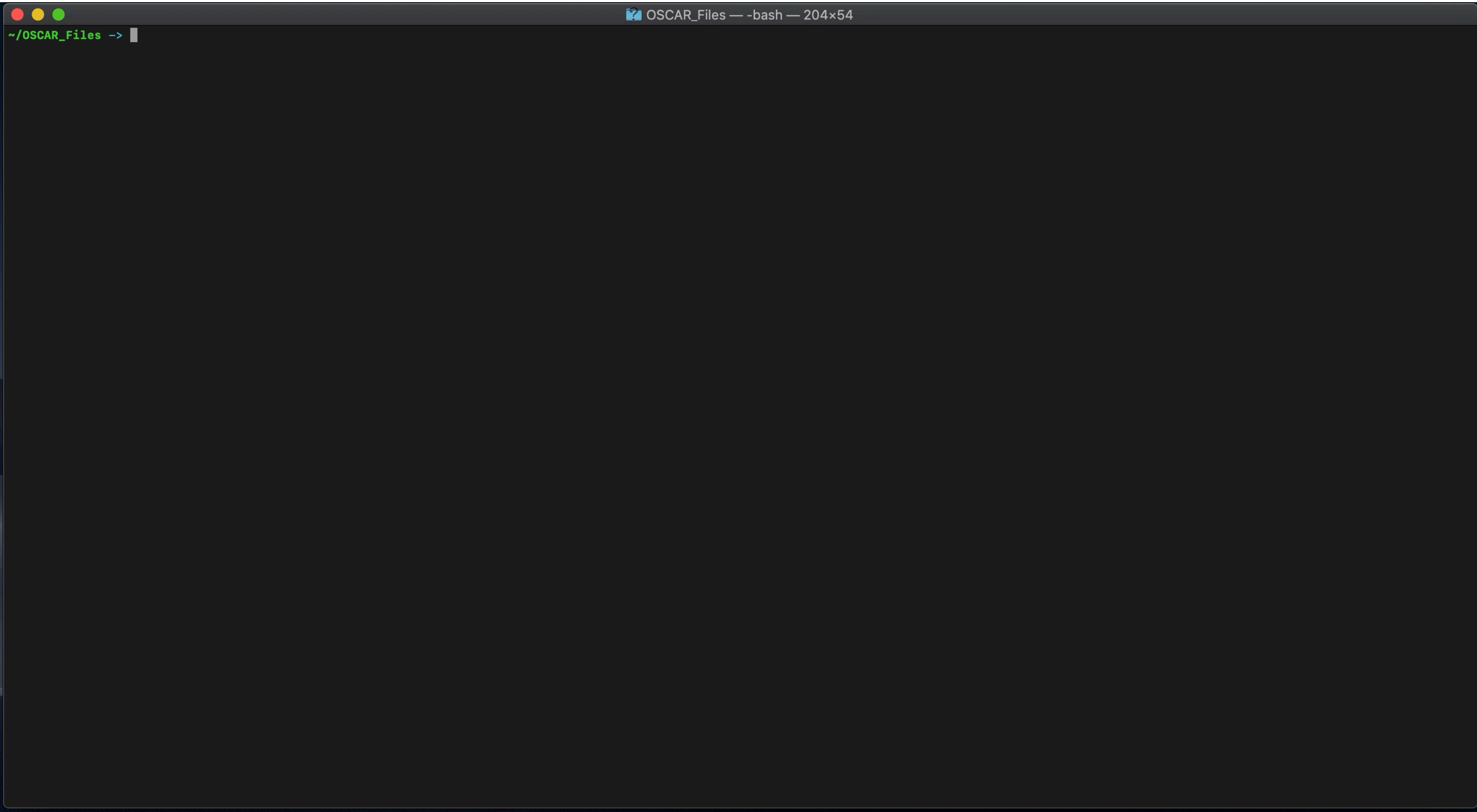
How could an automated reasoner have helped?

- Instantly detect the inconsistency
  - In this case, notice that Pilot 1's sensor reading seems unusual, and that Pilot 2's reading matches the backup instruments.
- Find a solution

How could an automated reasoner have helped?

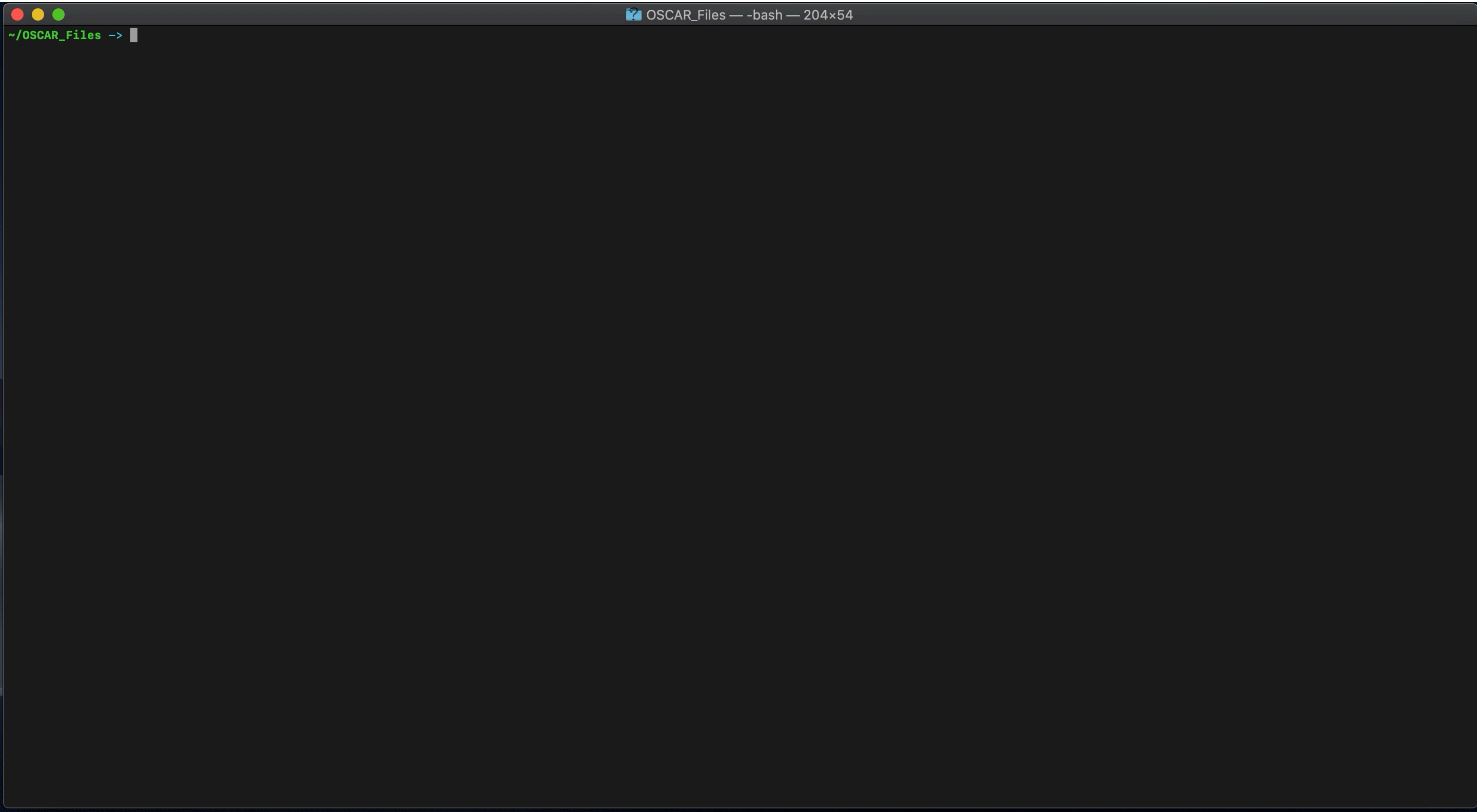
- Instantly detect the inconsistency
    - In this case, notice that Pilot 1's sensor reading seems unusual, and that Pilot 2's reading matches the backup instruments.
- Find a solution
    - In this case, send sensor readings from Pilot 2's sensors to Pilot 1's display, ignoring faulty data

A Solution in OSCAR

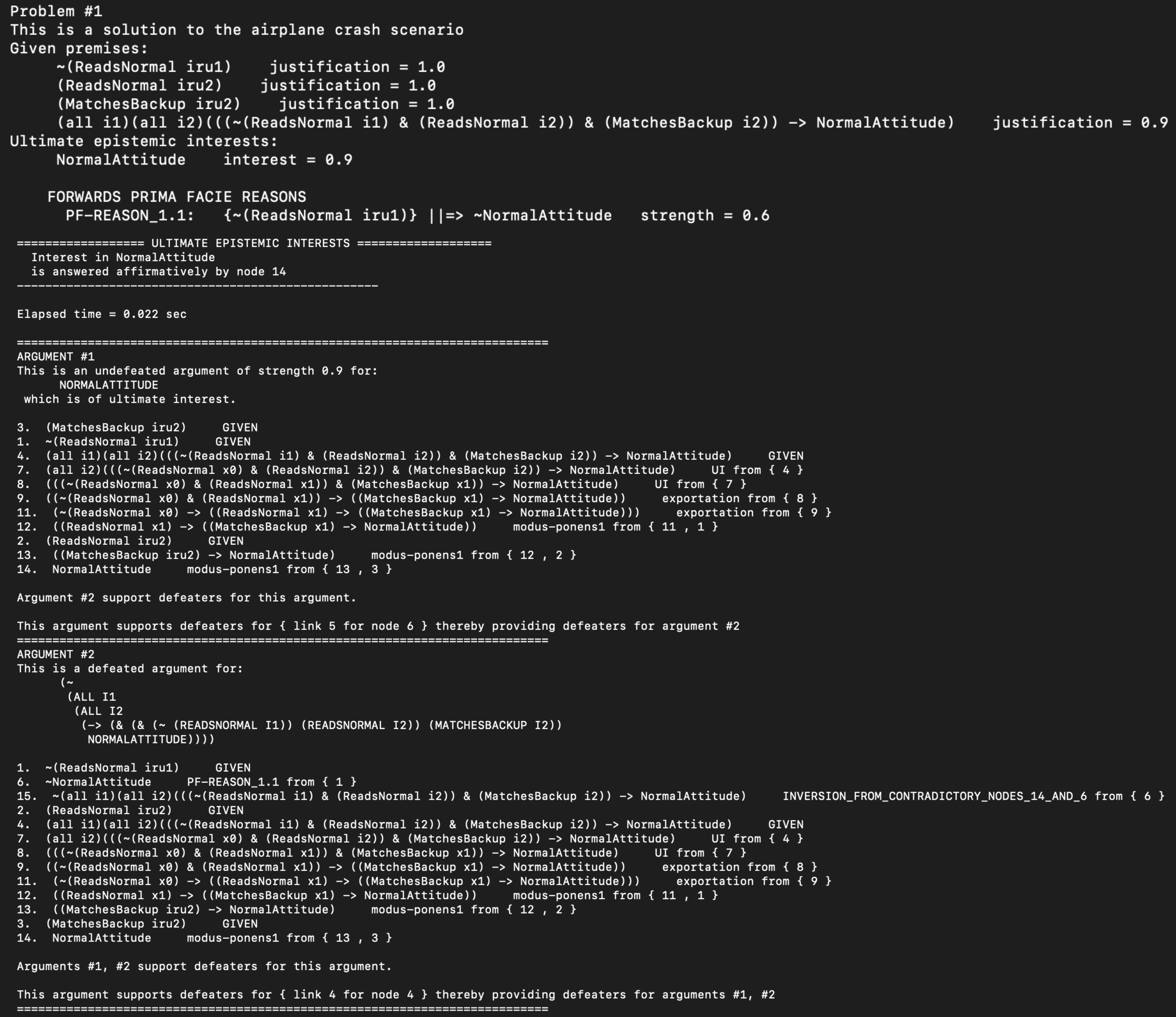


```
~/OSCAR_Files -> 
```

A Solution in OSCAR



```
~/OSCAR_Files ->
```



```
Problem #1
This is a solution to the airplane crash scenario
Given premises:
     ~(ReadsNormal iru1)    justification = 1.0
     (ReadsNormal iru2)    justification = 1.0
     (MatchesBackup iru2)    justification = 1.0
     (all i1)(all i2)(((~(ReadsNormal i1) & (ReadsNormal i2)) & (MatchesBackup i2)) -> NormalAttitude)    justification = 0.9
Ultimate epistemic interests:
     NormalAttitude    interest = 0.9

    FORWARDS PRIMA FACIE REASONS
      PF-REASON_1.1:   {~(ReadsNormal iru1)} ||=> ~NormalAttitude   strength = 0.6

 ================== ULTIMATE EPISTEMIC INTERESTS ===================
   Interest in NormalAttitude
   is answered affirmatively by node 14
 ---------------------------------------------------

 Elapsed time = 0.022 sec

 ===========================================================================
 ARGUMENT #1
 This is an undefeated argument of strength 0.9 for:
       NORMALATTITUDE
  which is of ultimate interest.

 3.  (MatchesBackup iru2)     GIVEN
 1.  ~(ReadsNormal iru1)     GIVEN
 4.  (all i1)(all i2)(((~(ReadsNormal i1) & (ReadsNormal i2)) & (MatchesBackup i2)) -> NormalAttitude)     GIVEN
 7.  (all i2)(((~(ReadsNormal x0) & (ReadsNormal i2)) & (MatchesBackup i2)) -> NormalAttitude)     UI from { 4 }
 8.  (((~(ReadsNormal x0) & (ReadsNormal x1)) & (MatchesBackup x1)) -> NormalAttitude)     UI from { 7 }
 9.  ((~(ReadsNormal x0) & (ReadsNormal x1)) -> ((MatchesBackup x1) -> NormalAttitude))     exportation from { 8 }
 11.  (~(ReadsNormal x0) -> ((ReadsNormal x1) -> ((MatchesBackup x1) -> NormalAttitude)))     exportation from { 9 }
 12.  ((ReadsNormal x1) -> ((MatchesBackup x1) -> NormalAttitude))     modus-ponens1 from { 11 , 1 }
 2.  (ReadsNormal iru2)     GIVEN
 13.  ((MatchesBackup iru2) -> NormalAttitude)     modus-ponens1 from { 12 , 2 }
 14.  NormalAttitude     modus-ponens1 from { 13 , 3 }

 Argument #2 support defeaters for this argument.

 This argument supports defeaters for { link 5 for node 6 } thereby providing defeaters for argument #2
 ===========================================================================
 ARGUMENT #2
 This is a defeated argument for:
       (~
        (ALL I1
         (ALL I2
          (-> (& (& (~ (READSNORMAL I1)) (READSNORMAL I2)) (MATCHESBACKUP I2))
           NORMALATTITUDE))))

 1.  ~(ReadsNormal iru1)     GIVEN
 6.  ~NormalAttitude     PF-REASON_1.1 from { 1 }
 15.  ~(all i1)(all i2)(((~(ReadsNormal i1) & (ReadsNormal i2)) & (MatchesBackup i2)) -> NormalAttitude)     INVERSION_FROM_CONTRADICTORY_NODES_14_AND_6 from { 6 }
 2.  (ReadsNormal iru2)     GIVEN
 4.  (all i1)(all i2)(((~(ReadsNormal i1) & (ReadsNormal i2)) & (MatchesBackup i2)) -> NormalAttitude)     GIVEN
 7.  (all i2)(((~(ReadsNormal x0) & (ReadsNormal i2)) & (MatchesBackup i2)) -> NormalAttitude)     UI from { 4 }
 8.  (((~(ReadsNormal x0) & (ReadsNormal x1)) & (MatchesBackup x1)) -> NormalAttitude)     UI from { 7 }
 9.  ((~(ReadsNormal x0) & (ReadsNormal x1)) -> ((MatchesBackup x1) -> NormalAttitude))     exportation from { 8 }
 11.  (~(ReadsNormal x0) -> ((ReadsNormal x1) -> ((MatchesBackup x1) -> NormalAttitude)))     exportation from { 9 }
 12.  ((ReadsNormal x1) -> ((MatchesBackup x1) -> NormalAttitude))     modus-ponens1 from { 11 , 1 }
 13.  ((MatchesBackup iru2) -> NormalAttitude)     modus-ponens1 from { 12 , 2 }
 3.  (MatchesBackup iru2)     GIVEN
 14.  NormalAttitude     modus-ponens1 from { 13 , 3 }

 Arguments #1, #2 support defeaters for this argument.

 This argument supports defeaters for { link 4 for node 4 } thereby providing defeaters for arguments #1, #2
 ===========================================================================
```

For More…

- https://rair.cogsci.rpi.edu/projects/automated-reasoners/oscar/
  - Software to run OSCAR
  - For files to run example from today, email me: mike.j.giancola@gmail.com.
- Licato, John. "Formalizing deceptive reasoning in breaking bad: Default reasoning in a doxastic logic." *2015 AAAI Fall Symposium Series*. 2015.
  - https://www.aaai.org/ocs/index.php/FSS/FSS15/paper/download/11669/11486