UCAAT 2013

22 – 24 October - Paris

Model-based test generation of aircraft traffic attack scenarios using ADS-B standard signals

Julien Botella (Smartesting)
Phong Cao (THALES)
Cédric Civeit (Thales Raytheon Systems)
Daniel Gidoin (THALES)
Fabien Peureux (FEMTO-ST /CNRS; Smartesting)

Agenda

- Context, motivation and key challenges
- MBT to generate attack scenarios for ADS-B
- Illustration of the end-to-end process on a simple example
- Conclusion and future work

Agenda

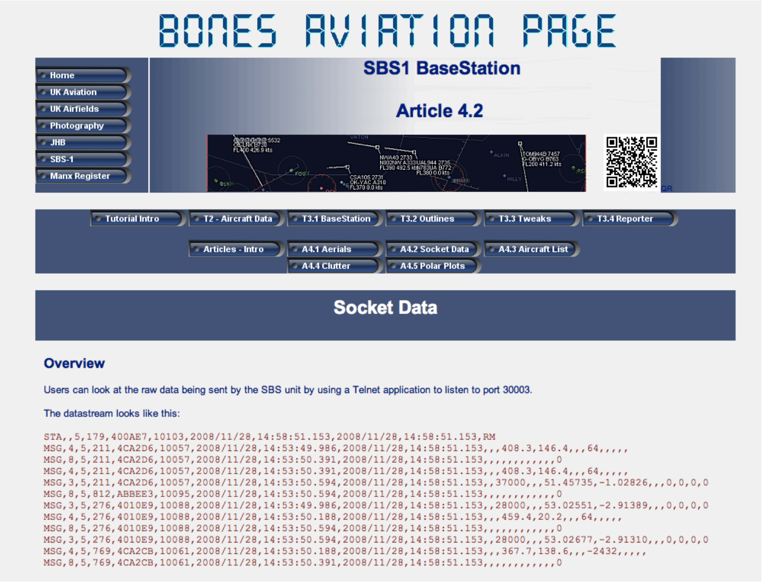
- Context, motivation and key challenges
- MBT to generate attack scenarios for ADS-B
- Illustration of the end-to-end process on a simple example
- Conclusion and future work

Automatic dependent surveillance-broadcast – ADS-B

- **Context**
  - To test air ADS-based Air Traffic Management systems using ADS-B Protocol
  - Radar control security testing:
    - ADS-B radio protocol
    - Flight information sent from plane to control tower
- **Motivations**
  - To address application security vulnerabilities that cannot be detected by the static tests
  - To reduce cost of testing and the time taken for industrialization
  - To be able to demonstrate the resilience of Air Traffic Management systems
  - To absorb the growth in air traffic and improve the security
- **Objectives**
  - Live traffic capture with SBS-3 station
  - Malicious scenario generation to check the detection efficiency from the control tower (logical anomalies)
    - Wrong coordinates
    - Fake planes
    - …
- **SBS-3 station description**

  http://www.homepages.mcb.net/bones/SBS/Article/Barebones42_Socket_Data.htm

SBS Specification extracts



BONES AVIATION PAGE

SBS1 BaseStation

Article 4.2

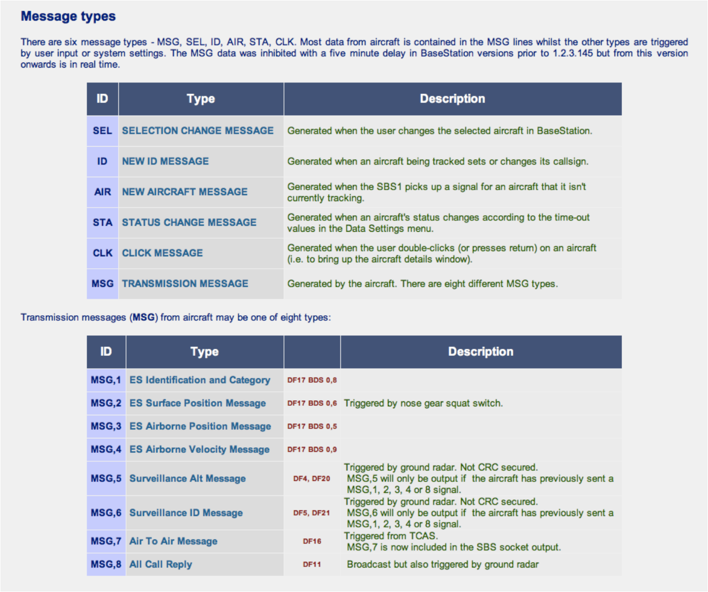
Socket Data

Overview

Users can look at the raw data being sent by the SBS unit by using a Telnet application to listen to port 30003.

The datastream looks like this:

```
STA,,5,179,400AE7,10103,2008/11/28,14:58:51.153,2008/11/28,14:58:51.153,RM
MSG,4,5,211,4CA2D6,10057,2008/11/28,14:53:49.986,2008/11/28,14:58:51.153,,,408.3,146.4,,,64,,,,,
MSG,8,5,211,4CA2D6,10057,2008/11/28,14:53:50.391,2008/11/28,14:58:51.153,,,,,,,,,,,,0
MSG,4,5,211,4CA2D6,10057,2008/11/28,14:53:50.391,2008/11/28,14:58:51.153,,,408.3,146.4,,,64,,,,,
MSG,3,5,211,4CA2D6,10057,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,37000,,,51.45735,-1.02826,,,0,0,0,0
MSG,8,5,812,ABBEE3,10095,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,,,,,,,,,,,0
MSG,3,5,276,4010E9,10088,2008/11/28,14:53:49.986,2008/11/28,14:58:51.153,,28000,,,53.02551,-2.91389,,,0,0,0,0
MSG,4,5,276,4010E9,10088,2008/11/28,14:53:50.188,2008/11/28,14:58:51.153,,,459.4,20.2,,,64,,,,,
MSG,8,5,276,4010E9,10088,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,,,,,,,,,,,0
MSG,3,5,276,4010E9,10088,2008/11/28,14:53:50.594,2008/11/28,14:58:51.153,,28000,,,53.02677,-2.91310,,,0,0,0,0
MSG,4,5,769,4CA2CB,10061,2008/11/28,14:53:50.188,2008/11/28,14:58:51.153,,,367.7,138.6,,,-2432,,,,,
MSG,8,5,769,4CA2CB,10061,2008/11/28,14:53:50.391,2008/11/28,14:58:51.153,,,,,,,,,,,,0
```



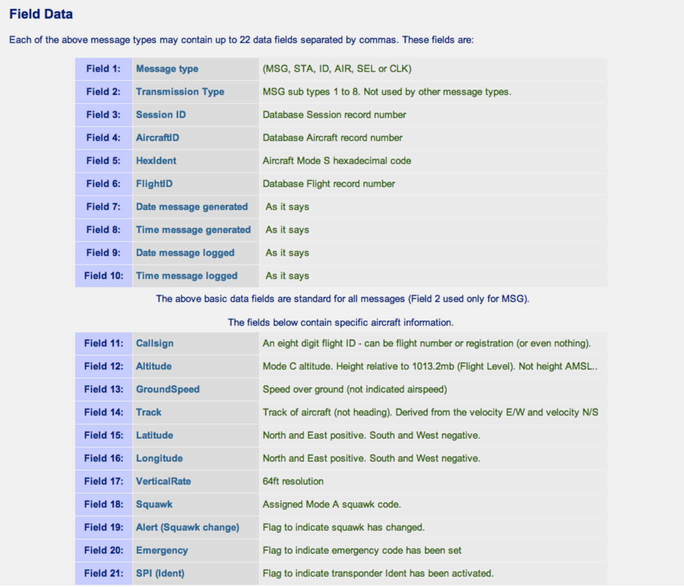
Message types

There are six message types - MSG, SEL, ID, AIR, STA, CLK. Most data from aircraft is contained in the MSG lines whilst the other types are triggered by user input or system settings. The MSG data was inhibited with a five minute delay in BaseStation versions prior to 1.2.3.145 but from this version onwards is in real time.

ID	Type	Description
SEL	SELECTION CHANGE MESSAGE	Generated when the user changes the selected aircraft in BaseStation.
ID	NEW ID MESSAGE	Generated when an aircraft being tracked sets or changes its callsign.
AIR	NEW AIRCRAFT MESSAGE	Generated when the SBS1 picks up a signal for an aircraft that it isn't currently tracking.
STA	STATUS CHANGE MESSAGE	Generated when an aircraft's status changes according to the time-out values in the Data Settings menu.
CLK	CLICK MESSAGE	Generated when the user double-clicks (or presses return) on an aircraft (i.e. to bring up the aircraft details window).
MSG	TRANSMISSION MESSAGE	Generated by the aircraft. There are eight different MSG types.

Transmission messages (**MSG**) from aircraft may be one of eight types:

ID	Type		Description
MSG,1	ES Identification and Category	DF17 BDS 0,8	
MSG,2	ES Surface Position Message	DF17 BDS 0,6	Triggered by nose gear squat switch.
MSG,3	ES Airborne Position Message	DF17 BDS 0,5	
MSG,4	ES Airborne Velocity Message	DF17 BDS 0,9	
MSG,5	Surveillance Alt Message	DF4, DF20	Triggered by ground radar. Not CRC secured. MSG,5 will only be output if the aircraft has previously sent a MSG,1, 2, 3, 4 or 8 signal.
MSG,6	Surveillance ID Message	DF5, DF21	Triggered by ground radar. Not CRC secured. MSG,6 will only be output if the aircraft has previously sent a MSG,1, 2, 3, 4 or 8 signal.
MSG,7	Air To Air Message	DF16	Triggered from TCAS. MSG,7 is now included in the SBS socket output.
MSG,8	All Call Reply	DF11	Broadcast but also triggered by ground radar



Field Data

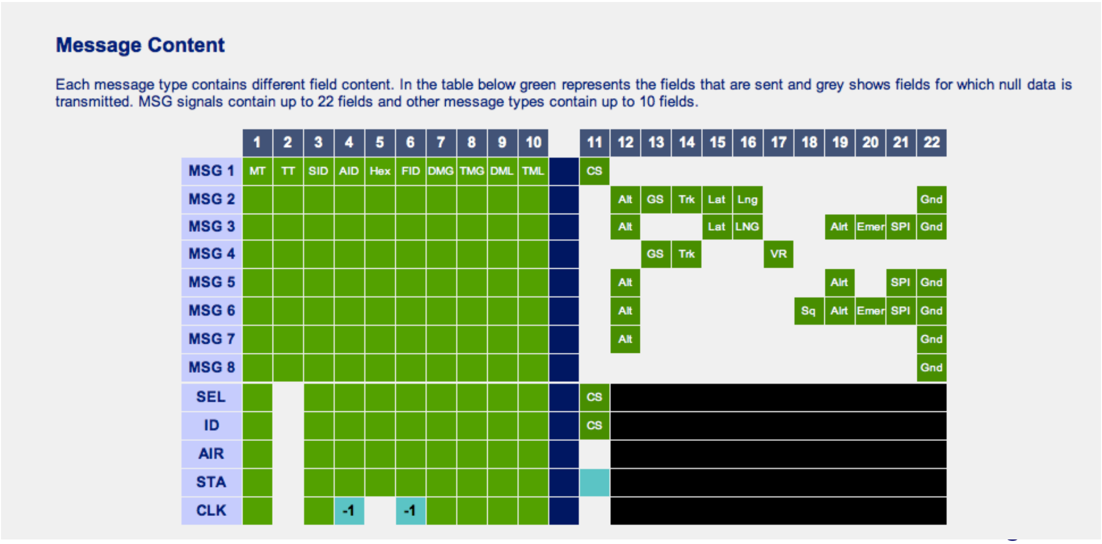
Each of the above message types may contain up to 22 data fields separated by commas. These fields are:

Field 1:	Message type	(MSG, STA, ID, AIR, SEL or CLK)
Field 2:	Transmission Type	MSG sub types 1 to 8. Not used by other message types.
Field 3:	Session ID	Database Session record number
Field 4:	AircraftID	Database Aircraft record number
Field 5:	HexIdent	Aircraft Mode S hexadecimal code
Field 6:	FlightID	Database Flight record number
Field 7:	Date message generated	As it says
Field 8:	Time message generated	As it says
Field 9:	Date message logged	As it says
Field 10:	Time message logged	As it says

The above basic data fields are standard for all messages (Field 2 used only for MSG).

The fields below contain specific aircraft information.

Field 11:	Callsign	An eight digit flight ID - can be flight number or registration (or even nothing).
Field 12:	Altitude	Mode C altitude. Height relative to 1013.2mb (Flight Level). Not height AMSL..
Field 13:	GroundSpeed	Speed over ground (not indicated airspeed)
Field 14:	Track	Track of aircraft (not heading). Derived from the velocity E/W and velocity N/S
Field 15:	Latitude	North and East positive. South and West negative.
Field 16:	Longitude	North and East positive. South and West negative.
Field 17:	VerticalRate	64ft resolution
Field 18:	Squawk	Assigned Mode A squawk code.
Field 19:	Alert (Squawk change)	Flag to indicate squawk has changed.
Field 20:	Emergency	Flag to indicate emergency code has been set
Field 21:	SPI (Ident)	Flag to indicate transponder Ident has been activated.



Message Content

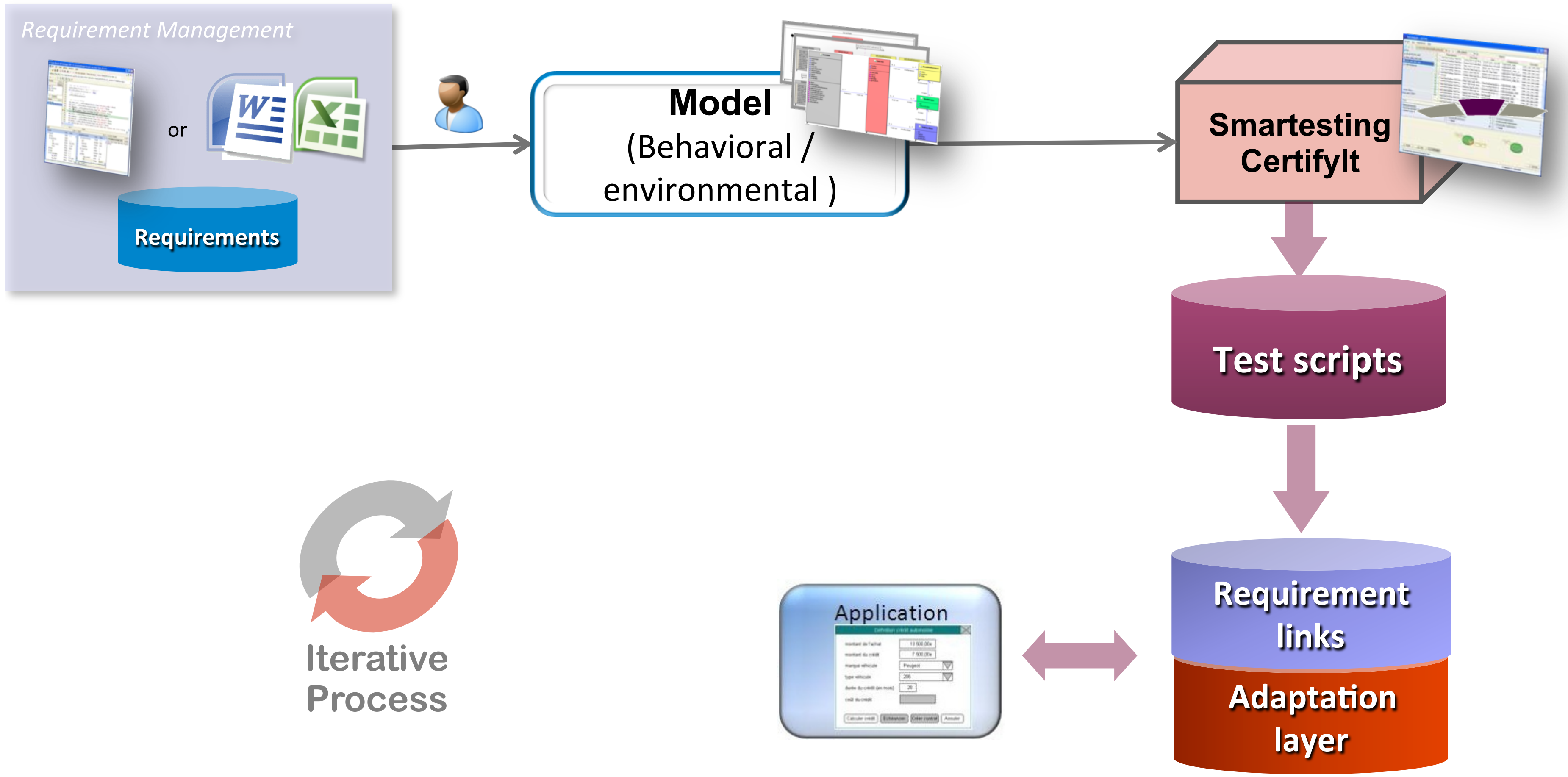
Each message type contains different field content. In the table below green represents the fields that are sent and grey shows fields for which null data is transmitted. MSG signals contain up to 22 fields and other message types contain up to 10 fields.

	1	2	3	4	5	6	7	8	9	10		11	12	13	14	15	16	17	18	19	20	21	22
MSG 1	MT	TT	SID	AID	Hex	FID	DMG	TMG	DML	TML		CS											
MSG 2	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			Alt	GS	Trk	Lat	Lng						Gnd
MSG 3	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			Alt			Lat	LNG			Alrt	Emer	SPI	Gnd
MSG 4	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩				GS	Trk			VR					
MSG 5	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			Alt							Alrt		SPI	Gnd
MSG 6	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			Alt						Sq	Alrt	Emer	SPI	Gnd
MSG 7	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			Alt										Gnd
MSG 8	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩													Gnd
SEL	🟩		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		CS	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
ID	🟩		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		CS	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
AIR	🟩		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
STA	🟩		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟦	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
CLK	🟩		🟩	-1		-1	🟩	🟩	🟩	🟩			⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛

Agenda

- Context, motivation and key challenges
- MBT to generate attack scenarios for ADS-B
- Illustration of the end-to-end process on a simple example
- Conclusion and future work

MBT for functional testing



Requirement Management

or

Requirements

Model
(Behavioral / environmental)

Smartesting CertifyIt

Test scripts

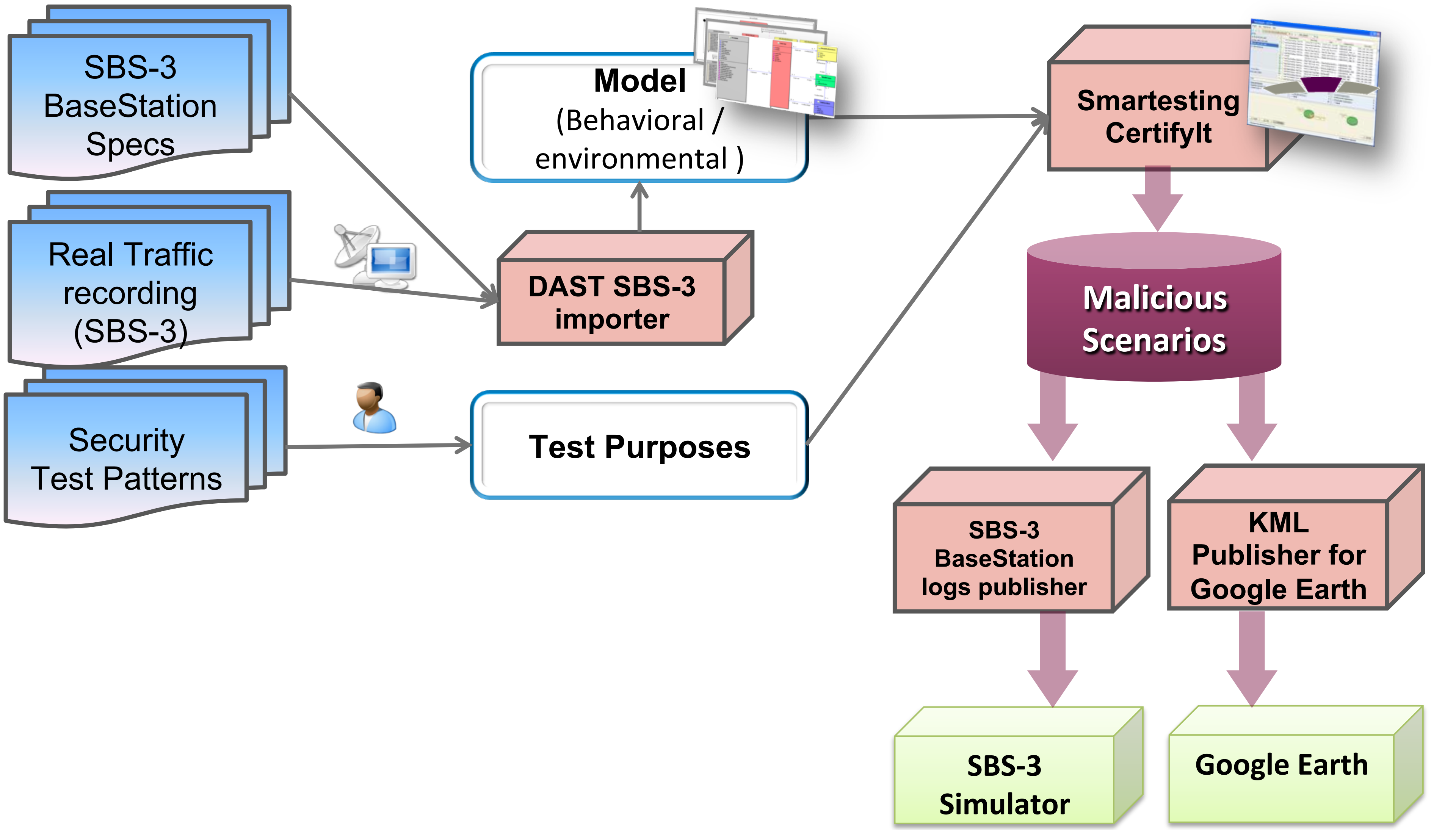
Requirement links

Adaptation layer

Application

Iterative Process

MBT process for ADS-B



SBS-3 BaseStation Specs

Real Traffic recording (SBS-3)

Security Test Patterns

Model (Behavioral / environmental)

DAST SBS-3 importer

Test Purposes

Smartesting CertifyIt

Malicious Scenarios

SBS-3 BaseStation logs publisher

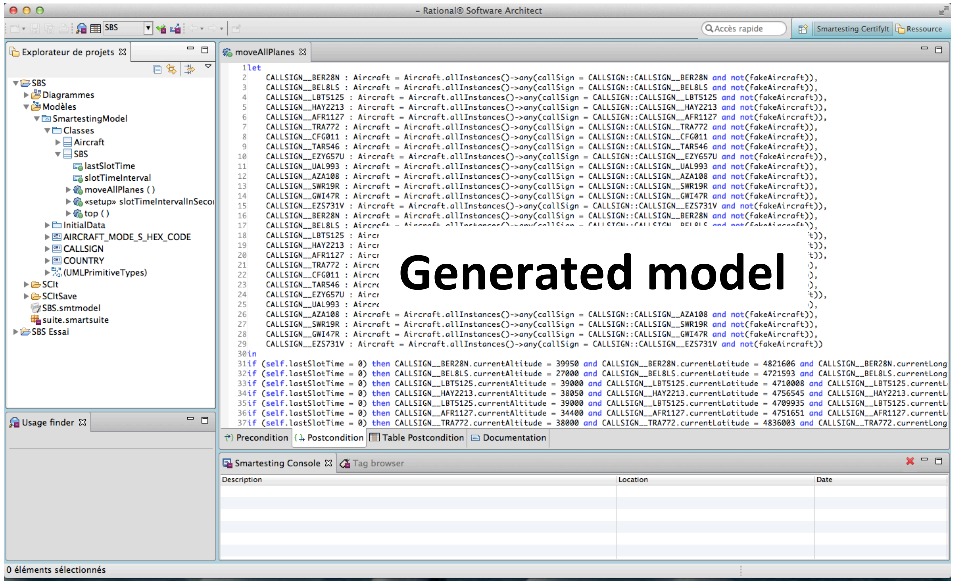
KML Publisher for Google Earth

SBS-3 Simulator

Google Earth

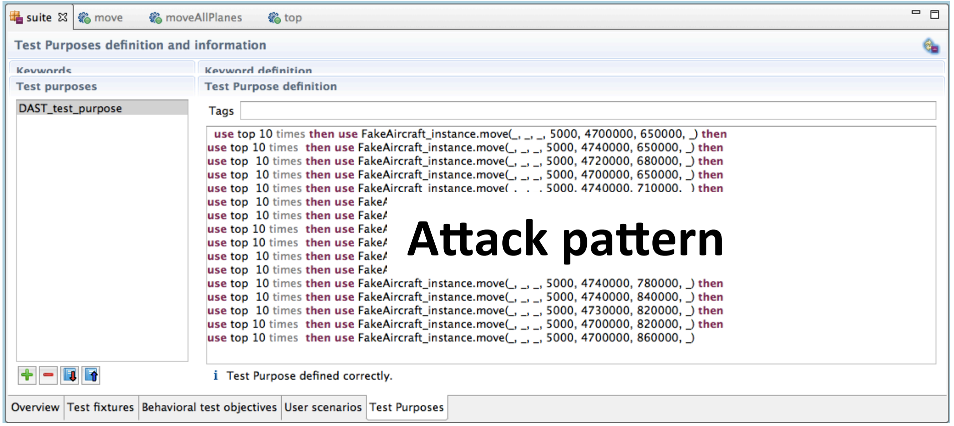
Test generation for ADS-B traces

- Attack scenarios are generated using real traces and attack patterns
- Attack patterns capture the know-how of security engineers



```
let
    CALLSIGN__BER28N : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__BER28N and not(fakeAircraft)),
    CALLSIGN__BEL8LS : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__BEL8LS and not(fakeAircraft)),
    CALLSIGN__LBT5125 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__LBT5125 and not(fakeAircraft)),
    CALLSIGN__HAY2213 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__HAY2213 and not(fakeAircraft)),
    CALLSIGN__AFR1127 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__AFR1127 and not(fakeAircraft)),
    CALLSIGN__TRA772 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__TRA772 and not(fakeAircraft)),
    CALLSIGN__CFG011 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__CFG011 and not(fakeAircraft)),
    CALLSIGN__TAR546 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__TAR546 and not(fakeAircraft)),
    CALLSIGN__EZY657U : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__EZY657U and not(fakeAircraft)),
    CALLSIGN__UAL993 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__UAL993 and not(fakeAircraft)),
    CALLSIGN__AZA108 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__AZA108 and not(fakeAircraft)),
    CALLSIGN__SWR19R : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__SWR19R and not(fakeAircraft)),
    CALLSIGN__GWI47R : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__GWI47R and not(fakeAircraft)),
    CALLSIGN__EZS731V : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__EZS731V and not(fakeAircraft)),
    CALLSIGN__BER28N : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__BER28N and not(fakeAircraft)),
    CALLSIGN__BEL8LS : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__BEL8LS and not(fakeAircraft)),
    CALLSIGN__LBT5125 : Aircr                                                                            t)),
    CALLSIGN__HAY2213 : Aircr                                                                            t)),
    CALLSIGN__AFR1127 : Aircr                                                                            t)),
    CALLSIGN__TRA772 : Aircra                                                                            ),
    CALLSIGN__CFG011 : Aircra                                                                            ),
    CALLSIGN__TAR546 : Aircra                                                                            ),
    CALLSIGN__EZY657U : Aircr                                                                            t)),
    CALLSIGN__UAL993 : Aircra                                                                            ),
    CALLSIGN__AZA108 : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__AZA108 and not(fakeAircraft)),
    CALLSIGN__SWR19R : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__SWR19R and not(fakeAircraft)),
    CALLSIGN__GWI47R : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__GWI47R and not(fakeAircraft)),
    CALLSIGN__EZS731V : Aircraft = Aircraft.allInstances()->any(callSign = CALLSIGN::CALLSIGN__EZS731V and not(fakeAircraft))
in
if (self.lastSlotTime = 0) then CALLSIGN__BER28N.currentAltitude = 39950 and CALLSIGN__BER28N.currentLatitude = 4821606 and CALLSIGN__BER28N.currentLong
if (self.lastSlotTime = 0) then CALLSIGN__BEL8LS.currentAltitude = 27000 and CALLSIGN__BEL8LS.currentLatitude = 4721593 and CALLSIGN__BEL8LS.currentLong
if (self.lastSlotTime = 0) then CALLSIGN__LBT5125.currentAltitude = 39000 and CALLSIGN__LBT5125.currentLatitude = 4710008 and CALLSIGN__LBT5125.currentL
if (self.lastSlotTime = 0) then CALLSIGN__HAY2213.currentAltitude = 38050 and CALLSIGN__HAY2213.currentLatitude = 4756545 and CALLSIGN__HAY2213.currentL
if (self.lastSlotTime = 0) then CALLSIGN__LBT5125.currentAltitude = 39000 and CALLSIGN__LBT5125.currentLatitude = 4709935 and CALLSIGN__LBT5125.currentL
if (self.lastSlotTime = 0) then CALLSIGN__AFR1127.currentAltitude = 34400 and CALLSIGN__AFR1127.currentLatitude = 4751651 and CALLSIGN__AFR1127.currentL
if (self.lastSlotTime = 0) then CALLSIGN__TRA772.currentAltitude = 38000 and CALLSIGN__TRA772.currentLatitude = 4836003 and CALLSIGN__TRA772.currentLong
```

Generated model



DAST_test_purpose

```
 use top 10 times then use FakeAircraft_instance.move(_, _, _, 5000, 4700000, 650000, _) then
use top 10 times  then use FakeAircraft_instance.move(_, _, _, 5000, 4740000, 650000, _) then
use top  10 times then use FakeAircraft_instance.move(_, _, _, 5000, 4720000, 680000, _) then
use top  10 times then use FakeAircraft_instance.move(_, _, _, 5000, 4700000, 650000, _) then
use top  10 times then use FakeAircraft instance.move( .  .  . 5000. 4740000. 710000.  ) then
use top  10 times then use FakeA
use top  10 times then use FakeA
use top 10 times  then use FakeA
use top 10 times  then use FakeA
use top  10 times then use FakeA
use top  10 times then use FakeA
use top  10 times then use FakeAircraft_instance.move(_, _, _, 5000, 4740000, 780000, _) then
use top  10 times then use FakeAircraft_instance.move(_, _, _, 5000, 4740000, 840000, _) then
use top  10 times then use FakeAircraft_instance.move(_, _, _, 5000, 4730000, 820000, _) then
use top 10 times  then use FakeAircraft_instance.move(_, _, _, 5000, 4700000, 820000, _) then
use top 10 times  then use FakeAircraft_instance.move(_, _, _, 5000, 4700000, 860000, _)
```

i Test Purpose defined correctly.

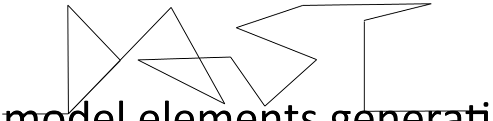
Attack pattern

Agenda

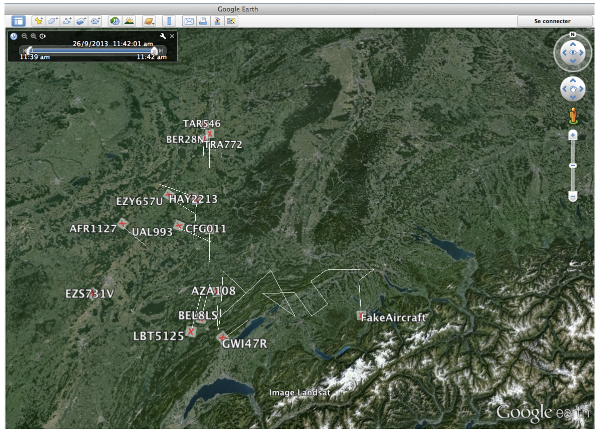
- Context, motivation and key challenges
- MBT to generate attack scenarios for ADS-B
- Illustration of the end-to-end process on a simple example (demo)
- Conclusion and future work

Project results

- **Goals**
  - To measure the resilience of Air Traffic Management Systems of against attacks using ADS_B protocol
  - The training of air traffic controllers in critical situations (i.e. artificial air space saturation)

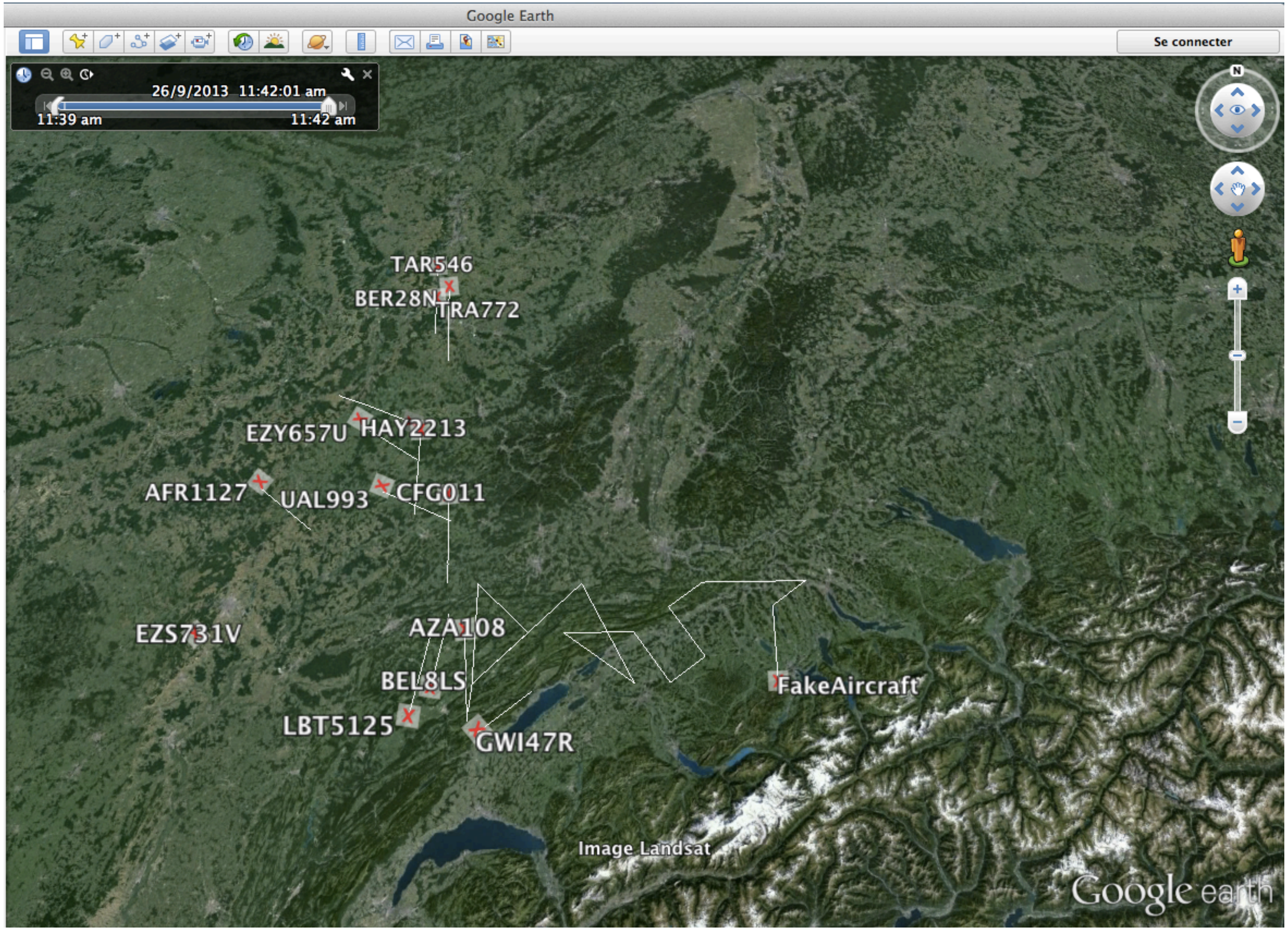


- **Process**
  - Automated real traffic acquisition (model elements generation)
  - Automatic malicious scenarios generation from test patterns
  - First pattern : DAST trajectory
  - Scenarios export (altered traffic)
    - KML forGoogle earth
    - SBS-3 formatted logs



- **Live Demo**

Simulating attack scenarios in Google Earth



Agenda

- Context, motivation and key challenges
- MBT to generate attack scenarios for ADS-B
- Illustration of the end-to-end process on a simple example (demo)
- Conclusion and future work

Future work

- Check injected data consistency
- Anomalie definitions to create new malicious scenarios
  - Vulnerability patterns (Q4 2013)
    - Fighter acting as an airliner
    - 4 grouped fighters, acting as an airliner then splitting
    - Helicopter, drone
    - Duplicate an airliner and make it diverge from its original trajectory
    - …
- KML/SBS exports improvements
- Improving tool integration (from generated attack scenarios to test execution, verdict and reporting)