Test and Evaluation of Cognitive EA systems - Requirements for future test systems

KEYSIGHT TECHNOLOGIES

Dan Pleasant, Solutions Architect, Keysight Technologies

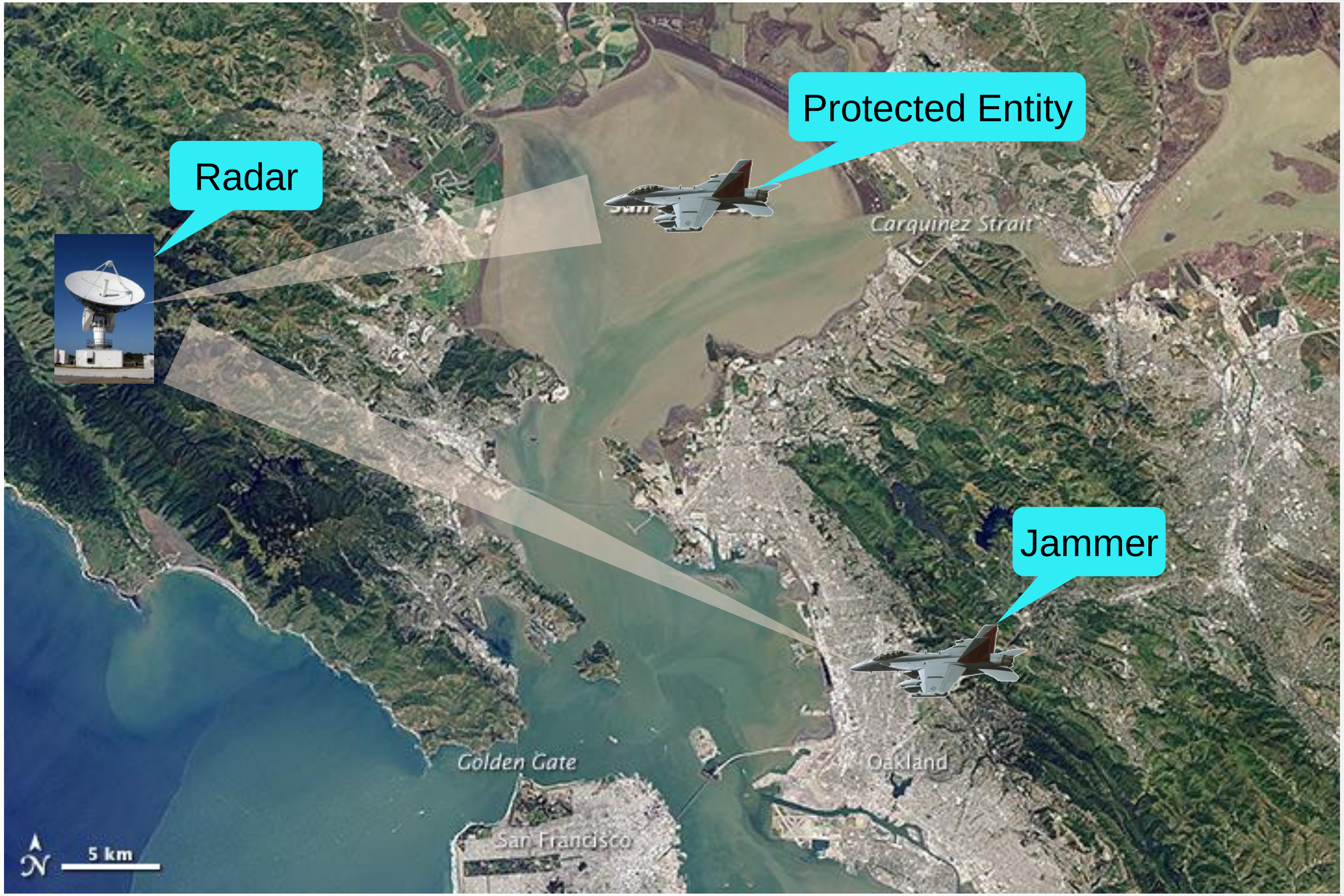
Situation

- You are designing and implementing cognitive electronic attack algorithms. How do you know that they will work in the field?
  - Computer simulations are a great start but sooner or later the algorithms must be tested in real hardware
  - Algorithms will need to be modified and improved many times, making flight range testing too expensive
- Need to test using hardware simulations of realistic conditions
- But… How do you test something if you don't know what it's going to do?
  - Cognitive algorithms mean the hardware will do whatever it wants - unpredictable
  - EA platform can maneuver autonomously
  - Need to emulate different kinds of radars, and they may be cognitive too
- How can you create an emulated environment?
- How can you tell if the EA algorithms are effective?

Simple Scenario

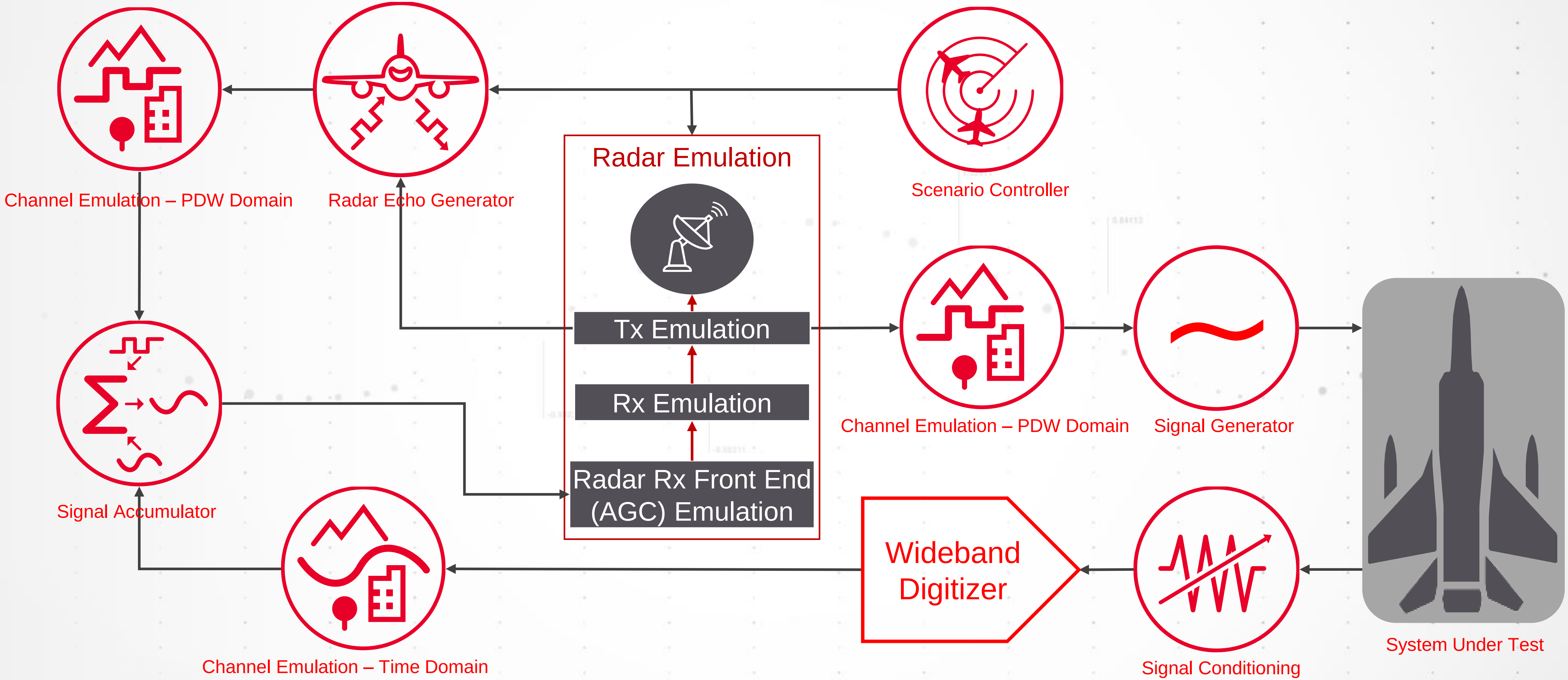
- Jammer is the SUT
- Everything else is emulated

- Scaling up to more complex scenarios to be considered later

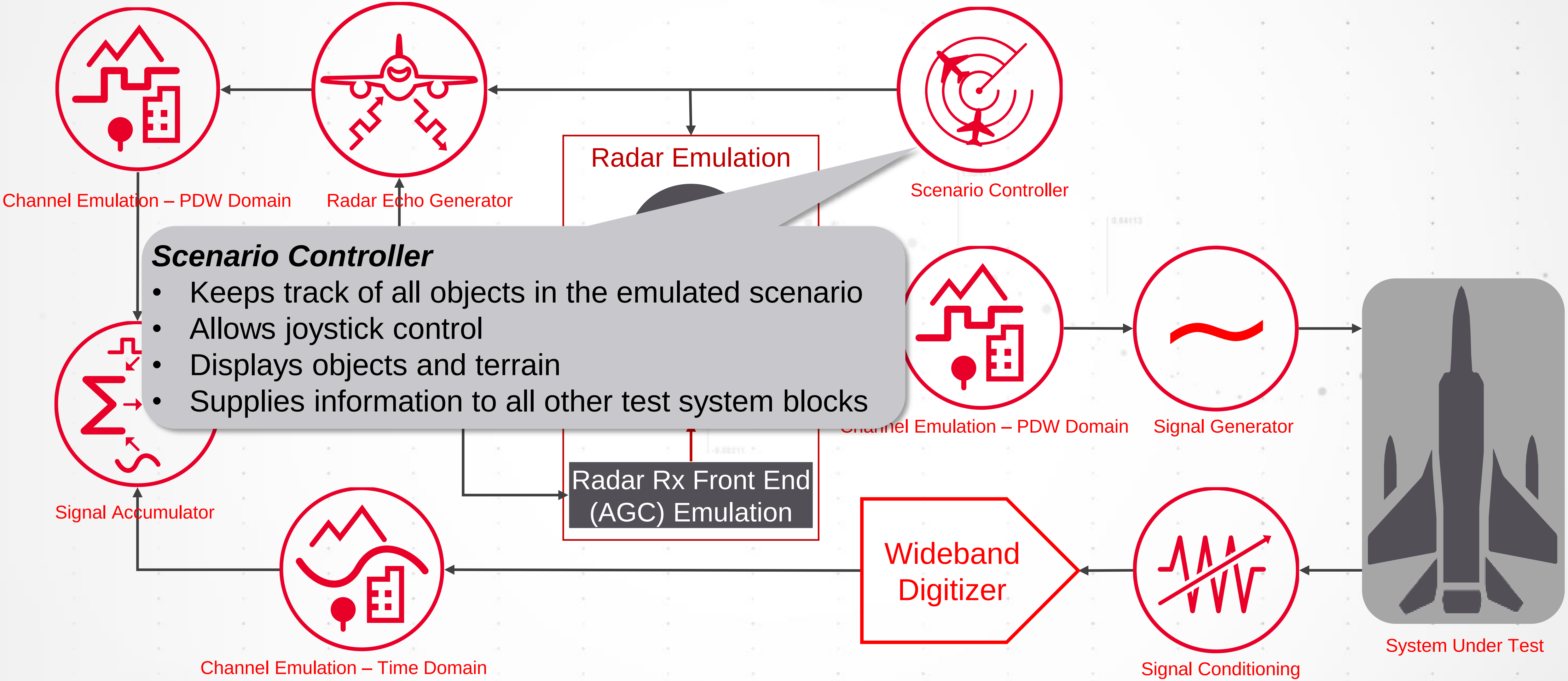


Landsat imagery courtesy of NASA Goddard Space Flight Center and U.S. Geological Survey

Test System Block Diagram



Test System Block Diagram



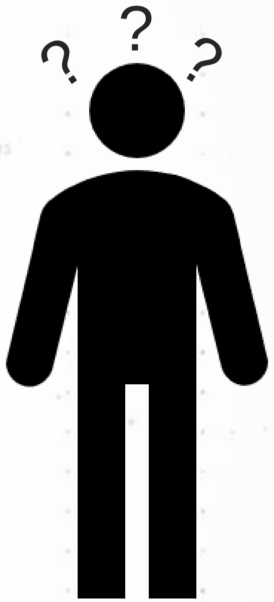
Scenario Controller

- Keeps track of all objects in the emulated scenario
- Allows joystick control
- Displays objects and terrain
- Supplies information to all other test system blocks

Challenges

SCENARIO CONTROLLER

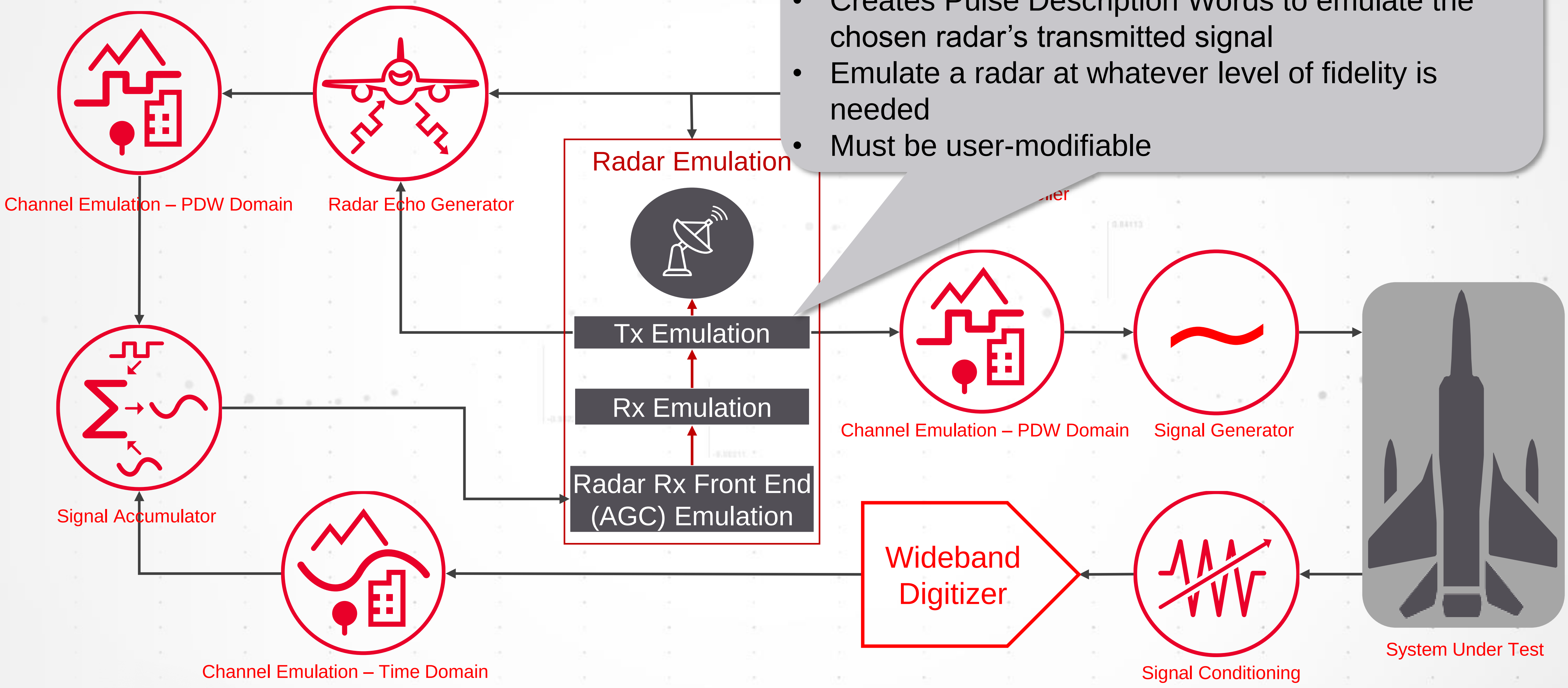
- SUT may function and maneuver autonomously
  - Need a two-way communication path between the SUT and the scenario controller
    - Scenario controller needs to know where the SUT is at any given time
    - SUT needs to know if it is about to crash into a mountain
- Timing is key – precise timing between all system elements is critical
- This scenario controller is not the same as the available COTS radar environment emulators.
  - Architected for closed-loop operation
  - Must have a high-speed interface that allows other test system blocks to obtain necessary data



Test System Block Diagram

Radar Transmitter Emulator

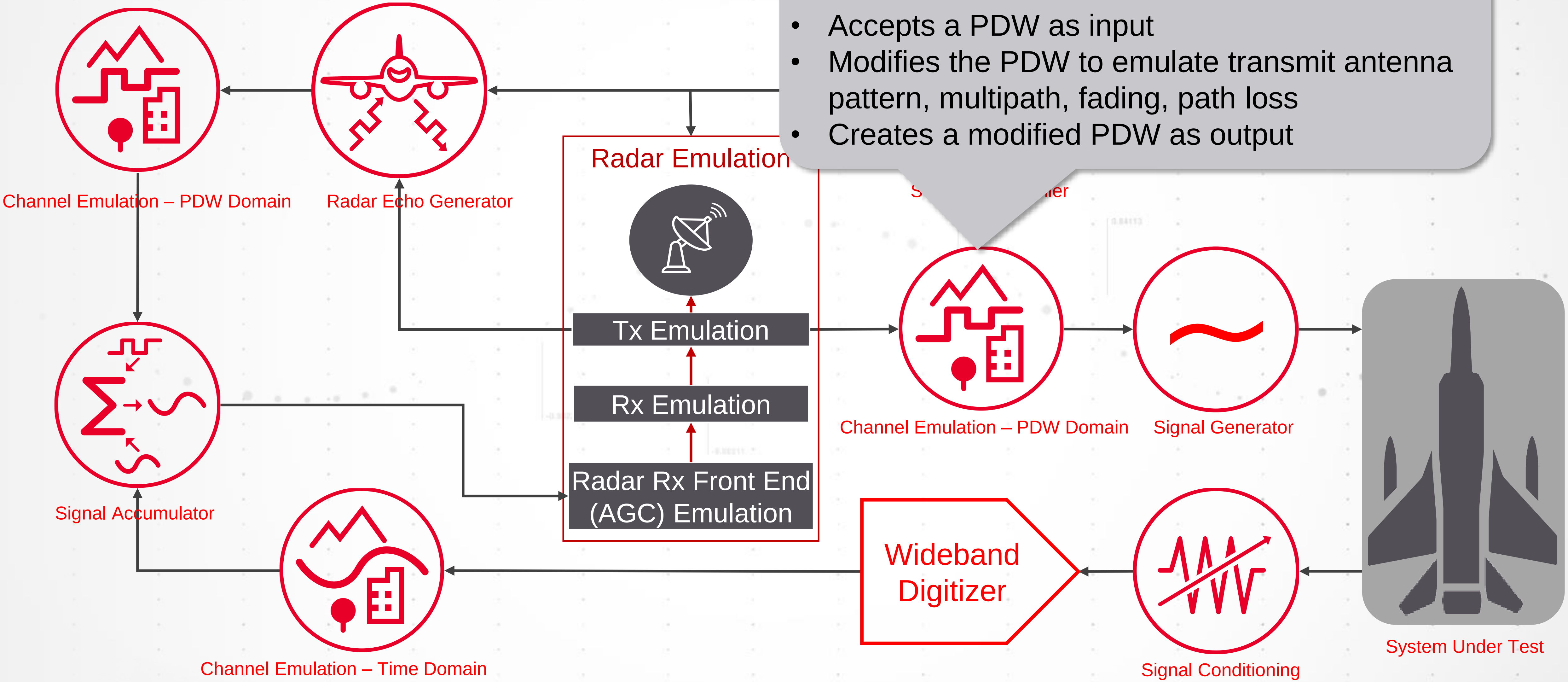
- Creates Pulse Description Words to emulate the chosen radar's transmitted signal
- Emulate a radar at whatever level of fidelity is needed
- Must be user-modifiable



Test System Block Diagram

Transmitter Channel Emulation

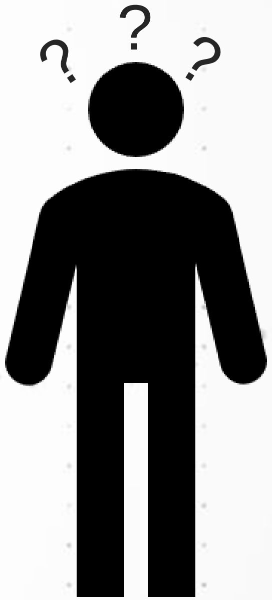
- Accepts a PDW as input
- Modifies the PDW to emulate transmit antenna pattern, multipath, fading, path loss
- Creates a modified PDW as output



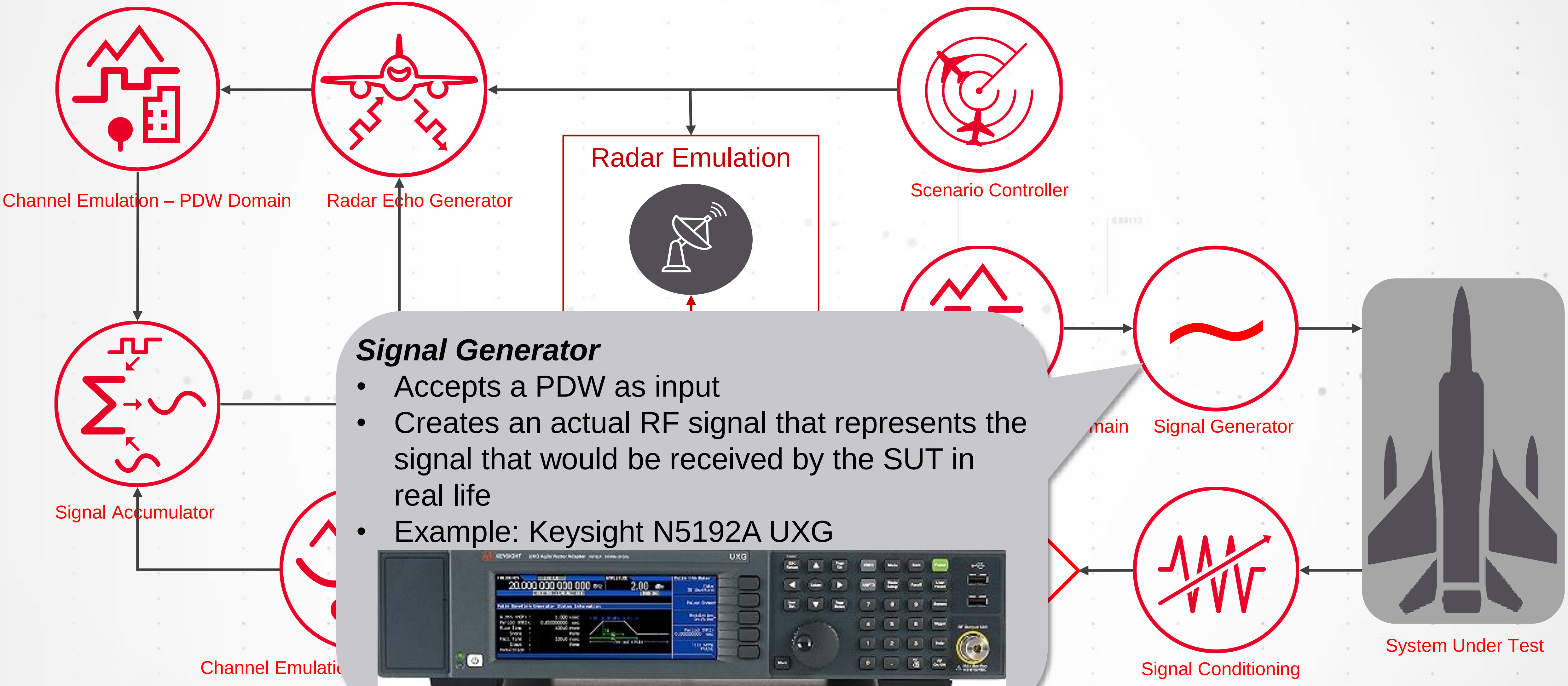
Challenges

TRANSMITTER CHANNEL EMULATOR

- Test system knows the beam direction of the radar but not the beam direction of the SUT
  - Need a communication path between the SUT and the transmit channel emulator in order to properly scale the SUT's input signal
- SUT antenna might be an array
  - Use many sources, or a beamforming network? Expensive
  - Insert transmitter signal behind the antenna? Need a physical connection point
- SUT might be trying to execute direction-finding algorithm using an antenna array
  - Expand the system to use many sources?
    - Requires a calibration step
    - Makes the system more expensive
  - Communicate with the SUT to 'push' the correct direction into the SUT's memory?
    - Requires special test mode and software communication with the SUT



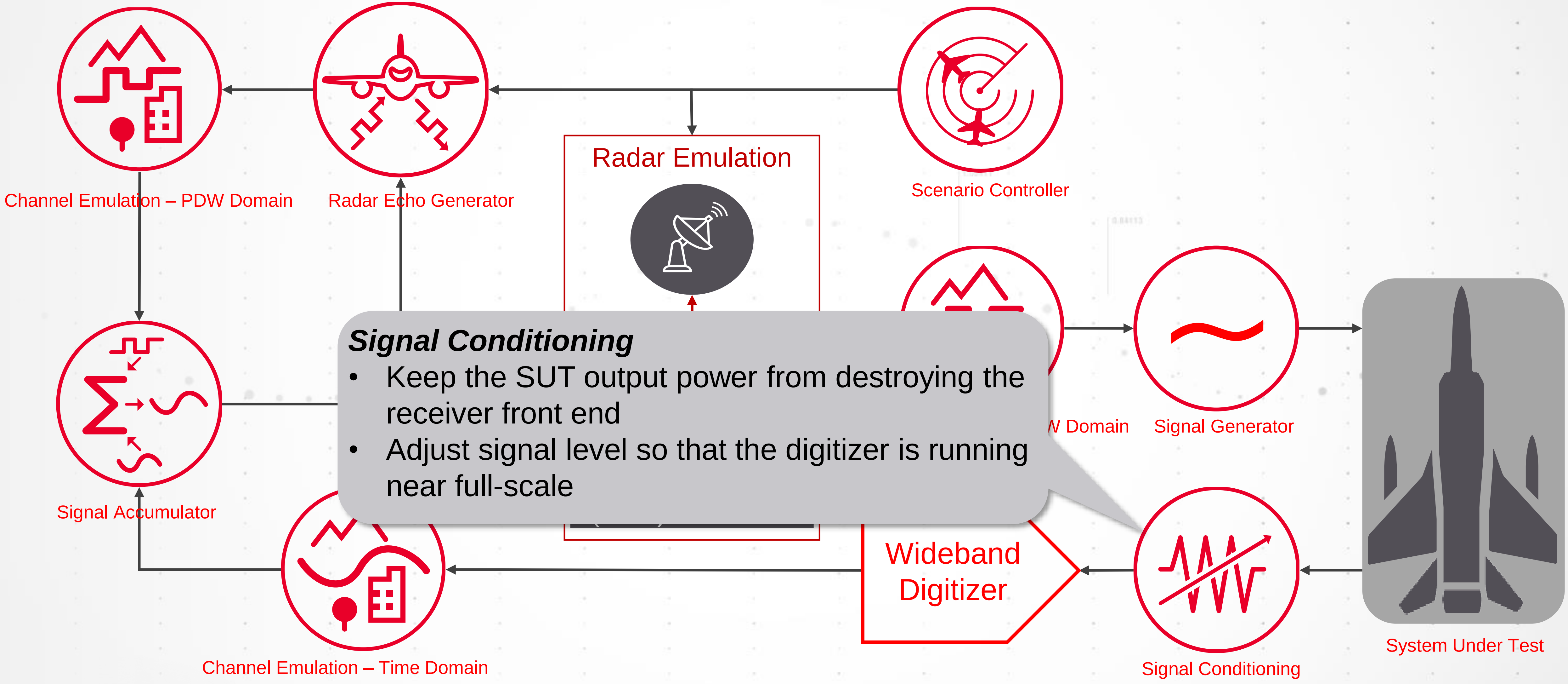
Test System Block Diagram



Signal Generator

- Accepts a PDW as input
- Creates an actual RF signal that represents the signal that would be received by the SUT in real life
- Example: Keysight N5192A UXG

Test System Block Diagram



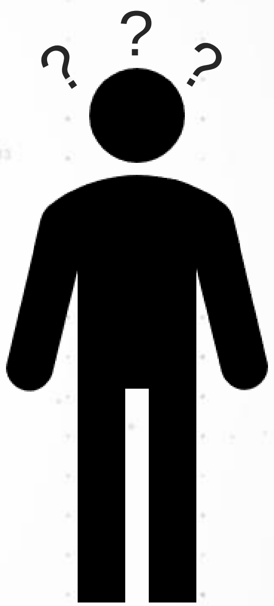
Signal Conditioning

- Keep the SUT output power from destroying the receiver front end
- Adjust signal level so that the digitizer is running near full-scale

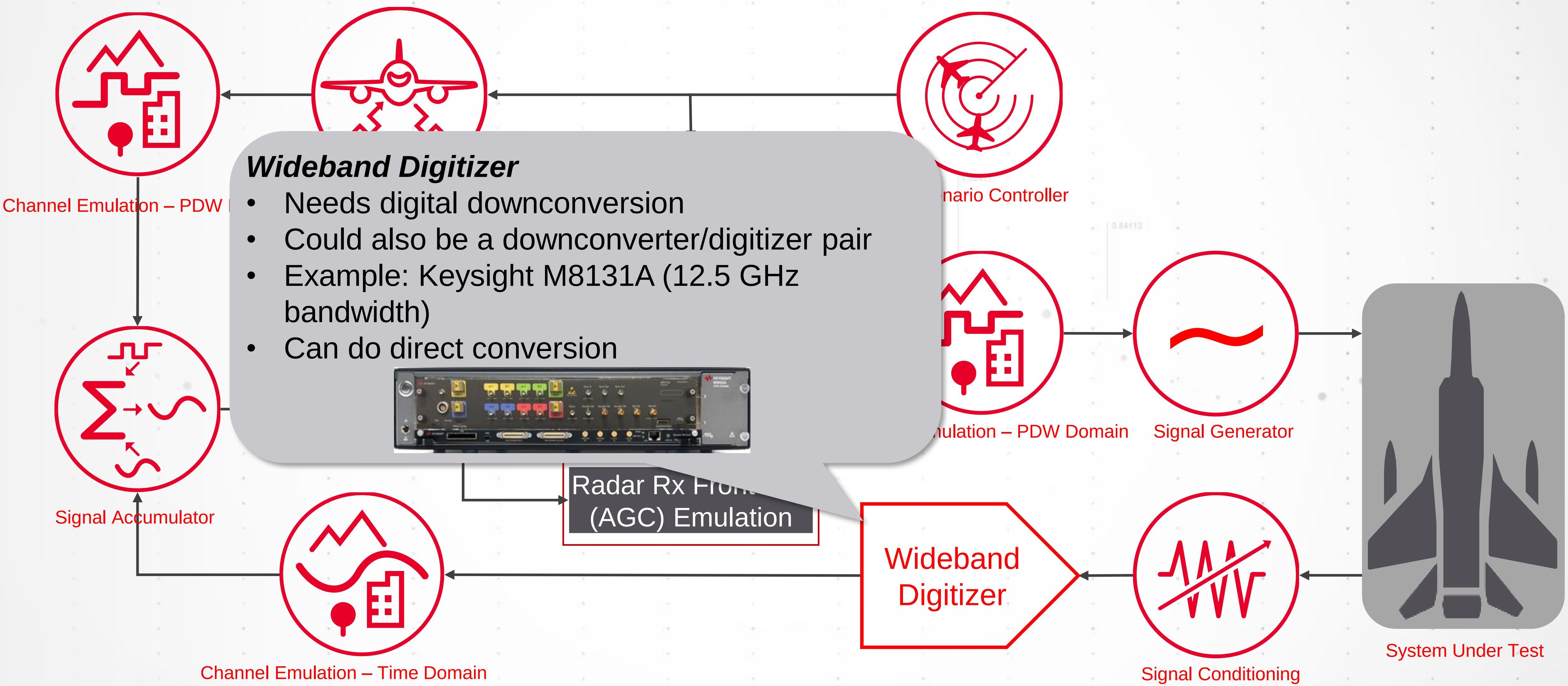
Challenges

SIGNAL CONDITIONING

- SUT output power many change rapidly at any time
  - Too-high power will damage the test system
  - Too-low power will result in loss of digitizer dynamic range
- Need SUT communication to predict power changes in advance
  - Need fast switching
- Or, implement very fast signal conditioning with power sensing
- Signal conditioning attenuation factor must be transmitted to other downstream test system blocks
  - Otherwise, radar emulator will think it always sees a high power signal



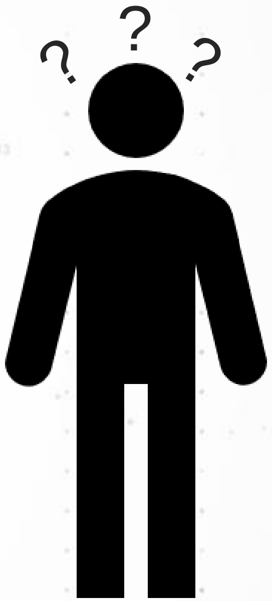
Test System Block Diagram



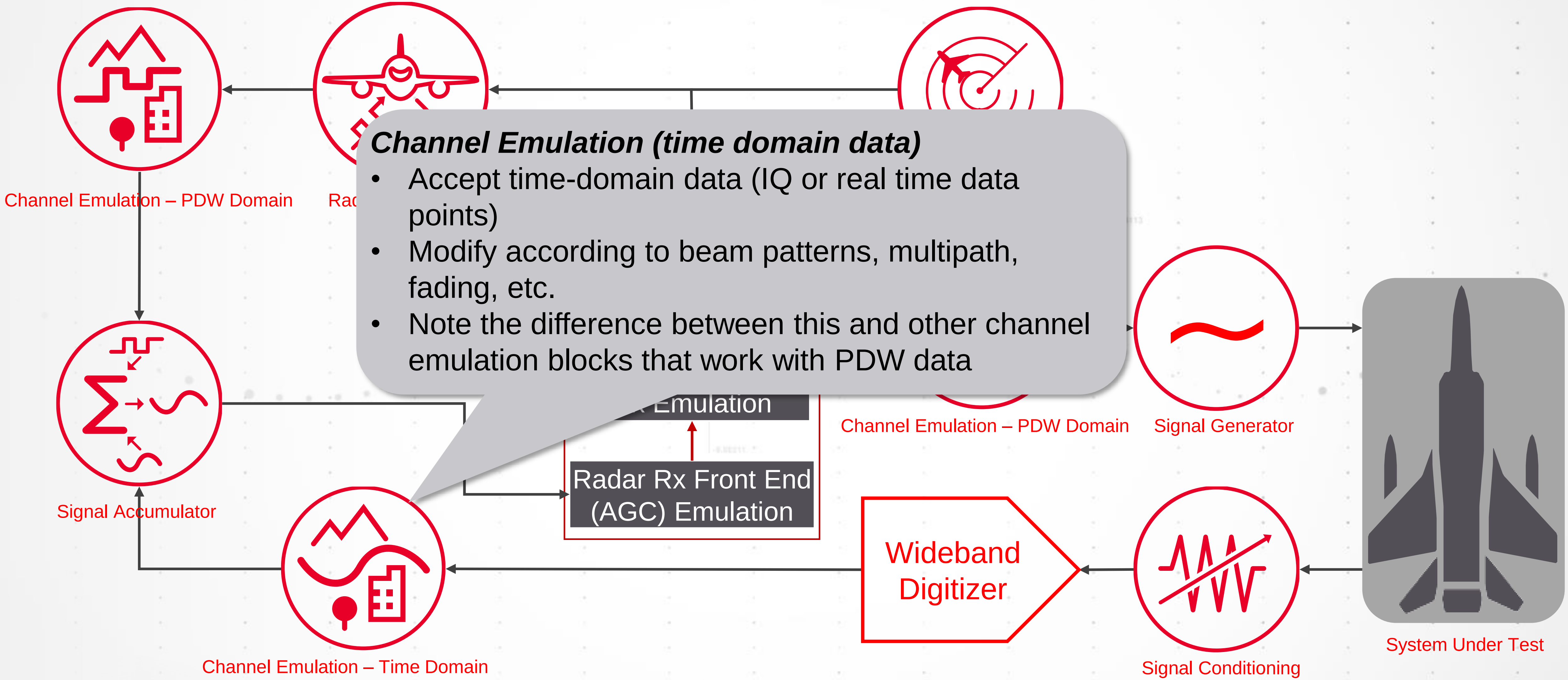
Challenge

DIGITIZER/RECEIVER

- SUT Signal may frequency-hop
  - Need to know the signal frequency to properly tune the receiver
- Need communication with SUT to tune properly
- Or, capture wide bandwidth and seek/select signals in DSP
- Or, query radar transmitter for the current frequency and assume the SUT will match that
  - Will not work if there is more than one radar in the scenario



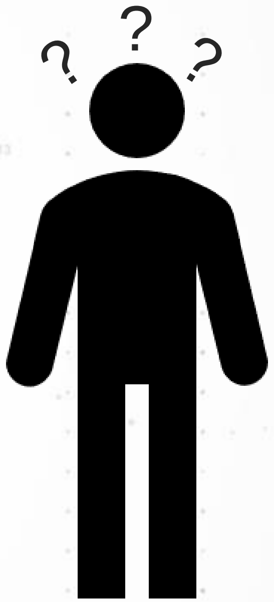
Test System Block Diagram



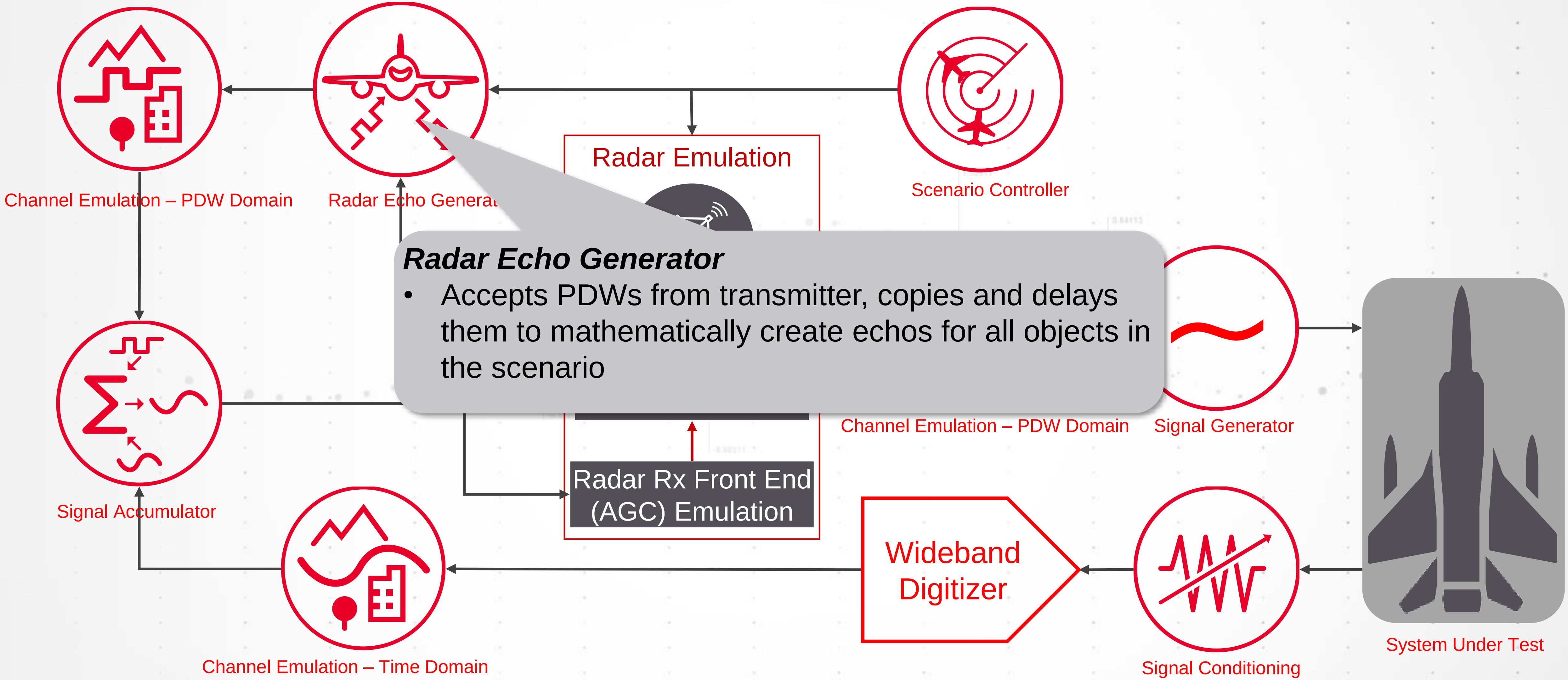
Challenges

RECEIVED SIGNAL CHANNEL EMULATION

- Test system does not know the beam-pointing direction of the SUT's transmit antenna
  - Required in order to correctly scale the received signal
  - Need communication with the SUT
- Channel emulation is computationally expensive
  - SUT may transmit any waveform, not necessarily just pulses
    - Channel emulator must work in time domain, not PDW domain
  - Available COTS channel emulators have sufficient bandwidth
    - …but usually take RF input, not digital, and only work up to 6 GHz or so



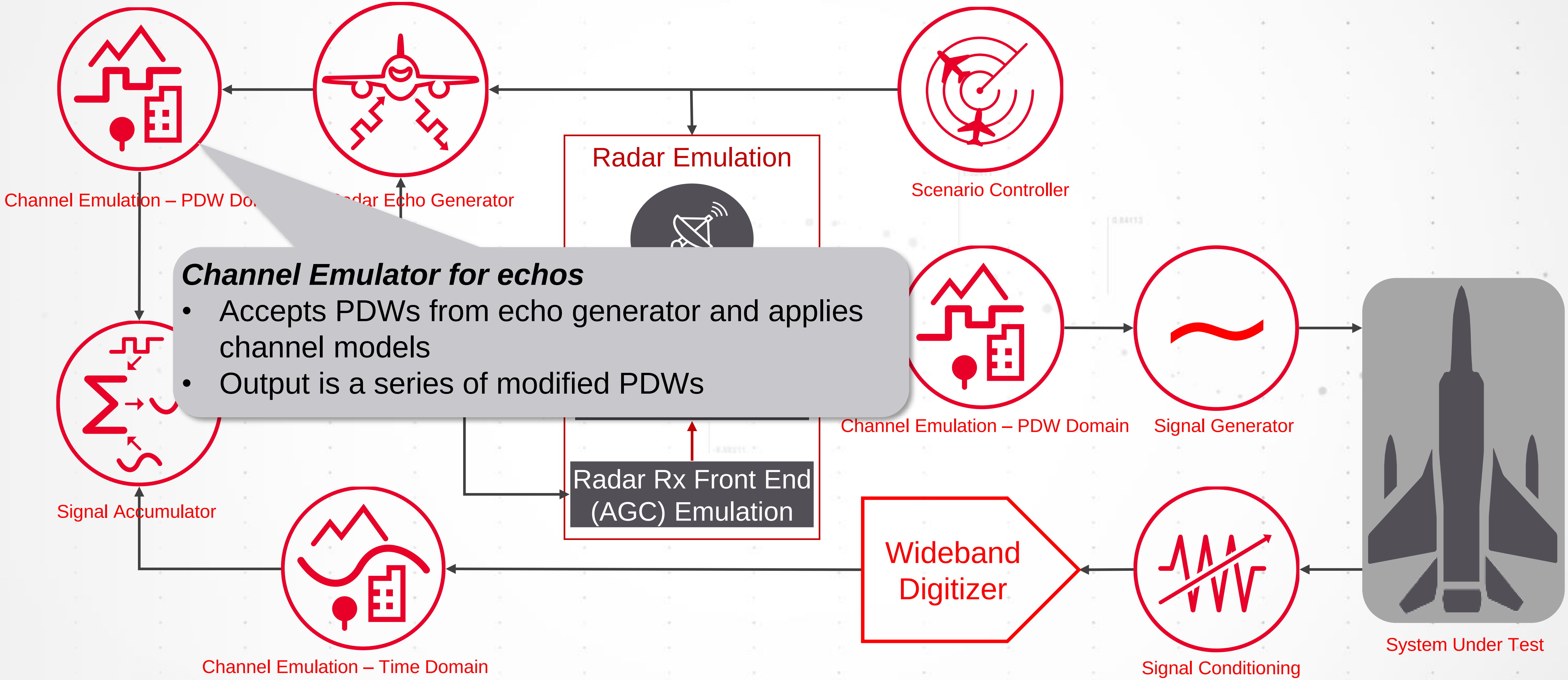
Test System Block Diagram



Radar Echo Generator

- Accepts PDWs from transmitter, copies and delays them to mathematically create echos for all objects in the scenario

Test System Block Diagram



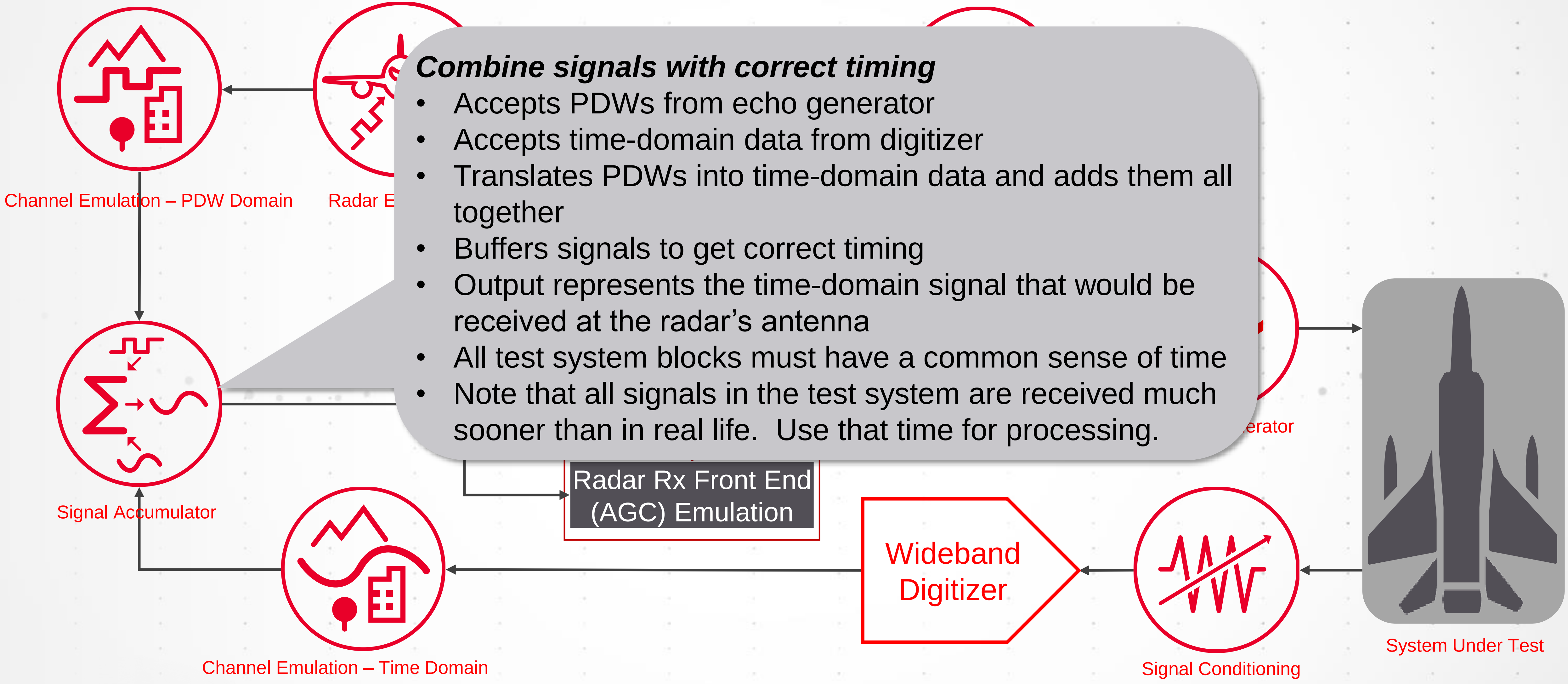
Channel Emulator for echos

- Accepts PDWs from echo generator and applies channel models
- Output is a series of modified PDWs

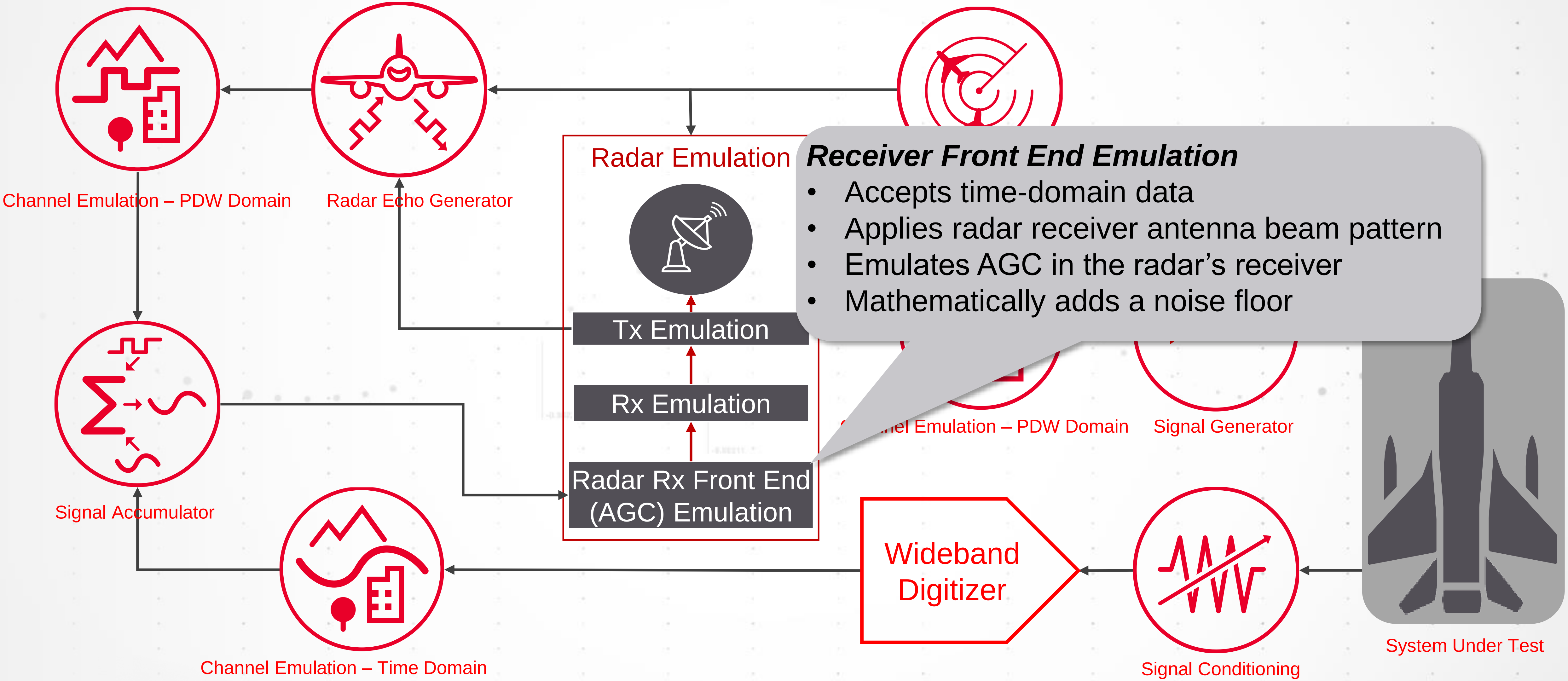
Test System Block Diagram

Combine signals with correct timing

- Accepts PDWs from echo generator
- Accepts time-domain data from digitizer
- Translates PDWs into time-domain data and adds them all together
- Buffers signals to get correct timing
- Output represents the time-domain signal that would be received at the radar's antenna
- All test system blocks must have a common sense of time
- Note that all signals in the test system are received much sooner than in real life. Use that time for processing.



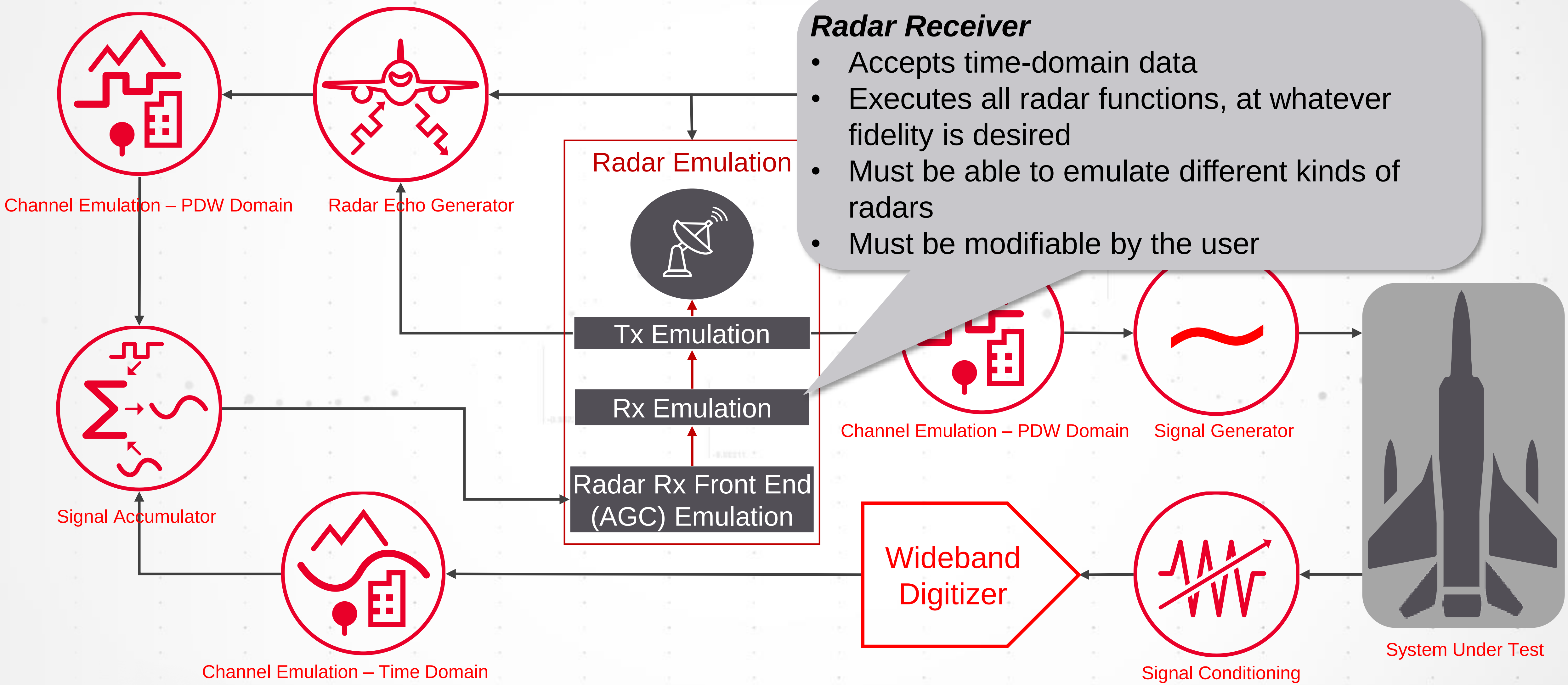
Test System Block Diagram



Receiver Front End Emulation

- Accepts time-domain data
- Applies radar receiver antenna beam pattern
- Emulates AGC in the radar's receiver
- Mathematically adds a noise floor

Test System Block Diagram



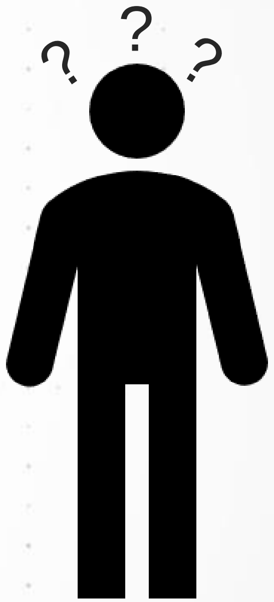
Radar Receiver

- Accepts time-domain data
- Executes all radar functions, at whatever fidelity is desired
- Must be able to emulate different kinds of radars
- Must be modifiable by the user

Challenges

RADAR RECEIVER

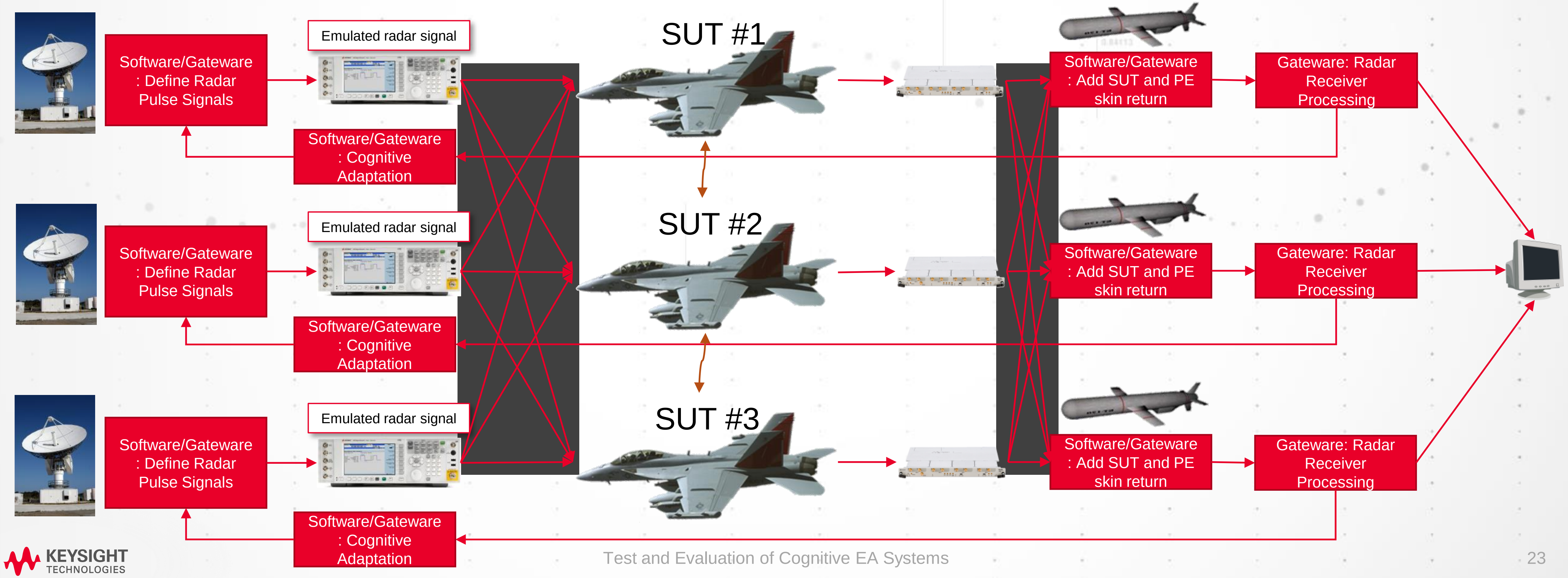
- Radar receiver algorithms must be modifiable by the user
  - Implies a library of FPGA-based IP that can be easily modified to implement different types of radars
  - Requires an application-specific tool that can be used to accomplish changes easily
    - Keysight PathWave:FPGA one example of such a tool
    - Still requires significant expertise on the part of the user
- There is no standard way to measure the SUT's effect
  - Success may be mission-dependent
    - Did the SUT successfully fool the radar for long enough?
    - Did the radar operator discover too quickly that the system was being attacked?
    - Was the radar able to get a fire-control solution on its target before the mission was complete?
  - Best current solution: Record all signals and carefully analyze later
    - Better test system would have real-time displays to ensure that everything is working
  - Industry would benefit from a standard, consistent measurement for EA effectiveness



Scalability

OBVIOUS SOLUTION IS CLUMSY

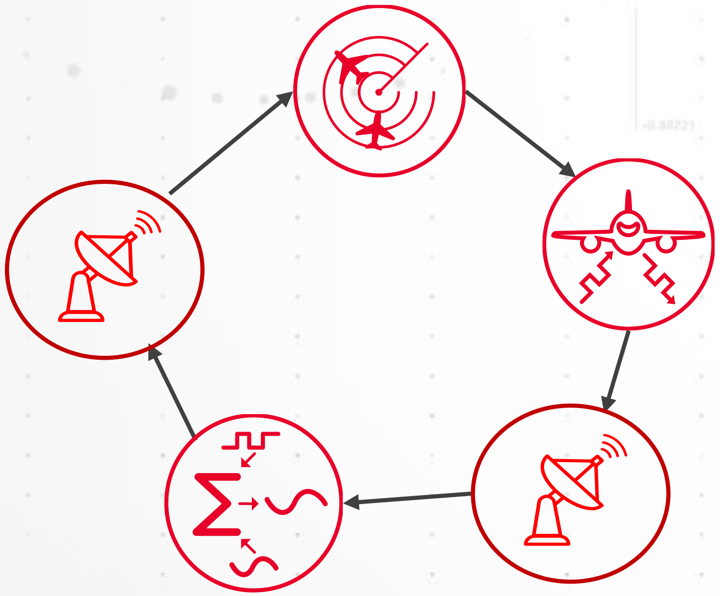
- Brute-force test system design is not scalable
  - Signal routing paths increase factorially with the number of objects in the scenario



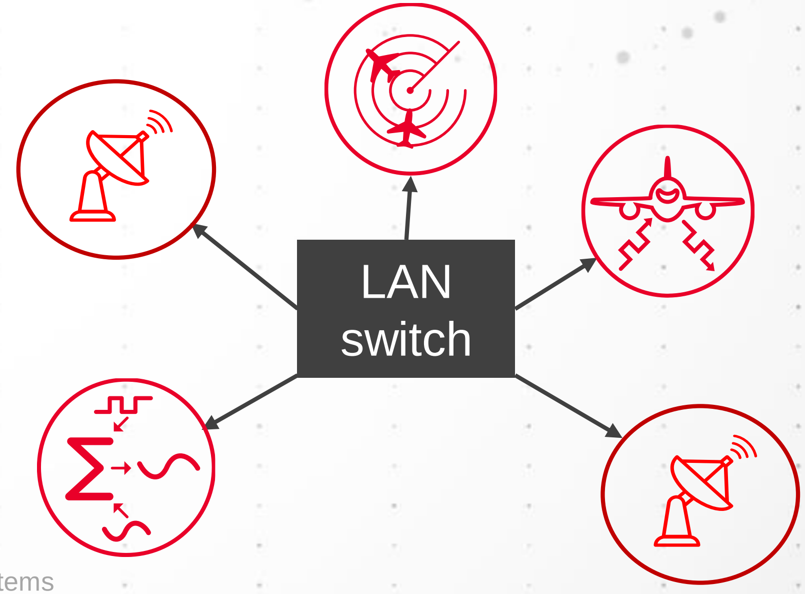
Scalability Solution? (But needs further investigation)

NOVEL TEST SYSTEM ARCHITECTURE POSSIBILITIES

- **Ring Architecture**
- All test system blocks form a ring network
- All data flows around the ring at high speed
- Test system blocks store the data they need, then forward everything to the next block



- **Star Architecture**
- Star configuration on a high-speed network
- All test system blocks broadcast relevant data to all other blocks
- Receiving blocks store the data they need, ignore the rest



Summary

THIS PROBLEM CAN BE SOLVED BUT NOT EASILY

- Cognitive EA system tests must be close-loop
  - Completely different architecture from common radar environment generators
- Significant communication with the SUT is strongly desired
  - Standard interface would be very welcome
- Measures of EA effectiveness have not been standardized
  - Industry-wide standard would be welcome
- Scalability is an issue
  - Requires new kinds of thinking regarding test system architecture
- **Call To Action**
  - The same kind of innovation needed in new operational systems (EA, comms, etc.) is also required for systems that test them
  - When architecting a new system, these things must be considered

KEYSIGHT
TECHNOLOGIES