



# Automated Display Testing in Test PASS

**May 30, 2019**

Confidential. Not to be copied, distributed, or reproduced without prior approval.





# Automated Display Testing in Test PASS

May 30, 2019

**Confidential. Not to be copied, reproduced, or distributed without prior approval.**

**CAUTION CONCERNING FORWARD-LOOKING STATEMENTS:**

This document contains "forward-looking statements" – that is, statements related to future events that by their nature address matters that are, to different degrees, uncertain. For details on the uncertainties that may cause our actual future results to be materially different than those expressed in our forward-looking statements, see <http://www.ge.com/investor-relations/disclaimer-caution-concerning-forwardlooking-statements> as well as our annual reports on Form 10-K and quarterly reports on Form 10-Q. We do not undertake to update our forward-looking statements. This document also includes certain forward-looking projected financial information that is based on current estimates and forecasts. Actual results could differ materially. to total risk-weighted assets.]

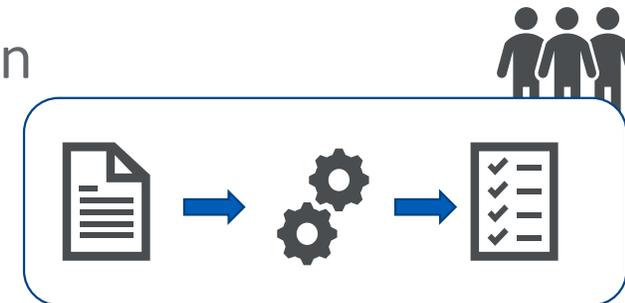
**NON-GAAP FINANCIAL MEASURES:**

In this document, we sometimes use information derived from consolidated financial data but not presented in our financial statements prepared in accordance with U.S. generally accepted accounting principles (GAAP). Certain of these data are considered "non-GAAP financial measures" under the U.S. Securities and Exchange Commission rules. These non-GAAP financial measures supplement our GAAP disclosures and should not be considered an alternative to the GAAP measure. The reasons we use these non-GAAP financial measures and the reconciliations to their most directly comparable GAAP financial measures are posted to the investor relations section of our website at [www.ge.com](http://www.ge.com). [We use non-GAAP financial measures including the following:

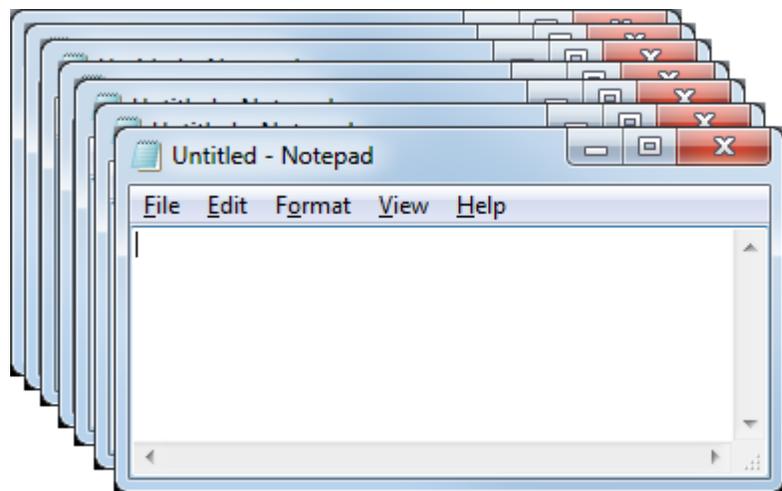
- Operating earnings and EPS, which is earnings from continuing operations excluding non-service-related pension costs of our principal pension plans.
- GE Industrial operating & Verticals earnings and EPS, which is operating earnings of our industrial businesses and the GE Capital businesses that we expect to retain.
- GE Industrial & Verticals revenues, which is revenue of our industrial businesses and the GE Capital businesses that we expect to retain.
- Industrial segment organic revenue, which is the sum of revenue from all of our industrial segments less the effects of acquisitions/dispositions and currency exchange.
- Industrial segment organic operating profit, which is the sum of segment profit from all of our industrial segments less the effects of acquisitions/dispositions and currency exchange.
- Industrial cash flows from operating activities (Industrial CFOA), which is GE's cash flow from operating activities excluding dividends received from GE Capital.
- Capital ending net investment (ENI), excluding liquidity, which is a measure we use to measure the size of our Capital segment.
- GE Capital Tier 1 Common ratio estimate is a ratio of equity

# Features of Test PASS

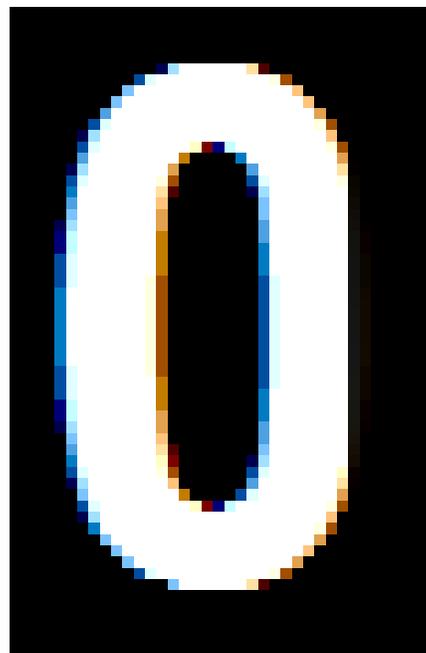
- Abbreviation: „Test Platform for Safety-Critical Systems”
- Integration of test environments
- Remote management and execution of tests
- Generation of unified report with rich information
- Test environment for display enhanced devices



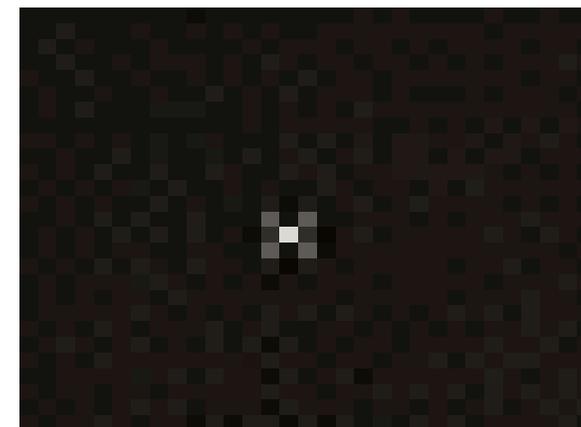
# Graphical errors of displays



Glitches



Cropping Occlusions



Dead



# Functional errors of displays

---

- Failure to satisfy requirements



SPEED 37

VS

*Requirement.*  
The displayed speed shall  
be lower than 35 knots.

# Automation

---

## Pros:

- Less human work
- Lower cost
- Greater precision
- Effective defect finders
- Repeatability

## Cons:

- Complex algorithms
- Machine Learning cannot be analyzed not fully tested
- Credibility issues

# Test PASS approach

---

- Automate reasoning about displayed content
- Simplify testing for displays
- Provide a language for description of user interface
- Move all advanced machine vision out of qualification rigor
- Qualify only a small piece of image processing code
- Express requirements with image semantics

# Problem to be solved

---

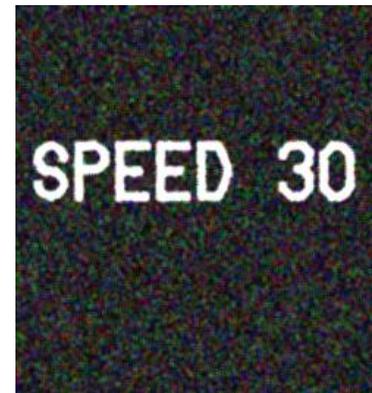
- How to eliminate a human from testing of displays?



*Requirement.*  
Displayed speed shall be  
lower than 35 knots.

# Verification by comparison

- Compare reference and captured image
- use industrial metric (e.g. MSE, SSIM) and apply thresholding



IMAGES ARE CONSISTENT TEST PASSED!

# Verification by comparison

---

- The only qualifiable part of image processing
- One takes credit only if comparison algorithms assesses that images are consistent
- Reducing number of false positives improves reliability of the tool but does not affect correctness

# Verification by comparison

---

## Issues:

- Separate image for each testcase => expensive test development
- Cannot handle undefined parts of image (except simple masking)
- Cannot express requirements like „Speed shall be lower than 35 knots”

# Parametrized models (p-models)

---

- Description of an image
- Human readable
- XML based
- Structured
- Renderable
- Stateless



Project: „An innovative tool for automated software testing for safety critical systems in aviation as a confirmation of the competences of Polish scientific staff of WIA on the international arena”. Founding agreement with NCBIR, Programme: Smart Growth Operational Programme 2014-2020, No: POIR.04.01.04-00-0121/16 signed 04/13/2017



# P-Models - rendering

```
<svg width="192" height="192"
xmlns="http://www.w3.org/2000/svg">

<rect width="100%" height="100%" fill="black"/>

<text id="speed"
font-size="30"
fill="white"
x="6" y="96"
>SPEED 30</text>

</svg>
```



# P-Models - reasoning

```
<svg width="192" height="192"  
xmlns="http://www.w3.org/2000/svg">
```

```
<rect width="100%" height="100%" fill="black"/>
```

```
<text id="speed"  
font-size="30"  
fill="white"  
x="6" y="96"  
>SPEED 30</text>
```

```
</svg>
```



```
# Formal specification of requirement  
speed = int(model.speed.text[-2:])  
Assert(speed < 35)
```



# P-model verification – key idea

---

- If
- The model is a correct interpretation of captured image and
- The model satisfies requirements
- Then
- Captured image satisfies requirements



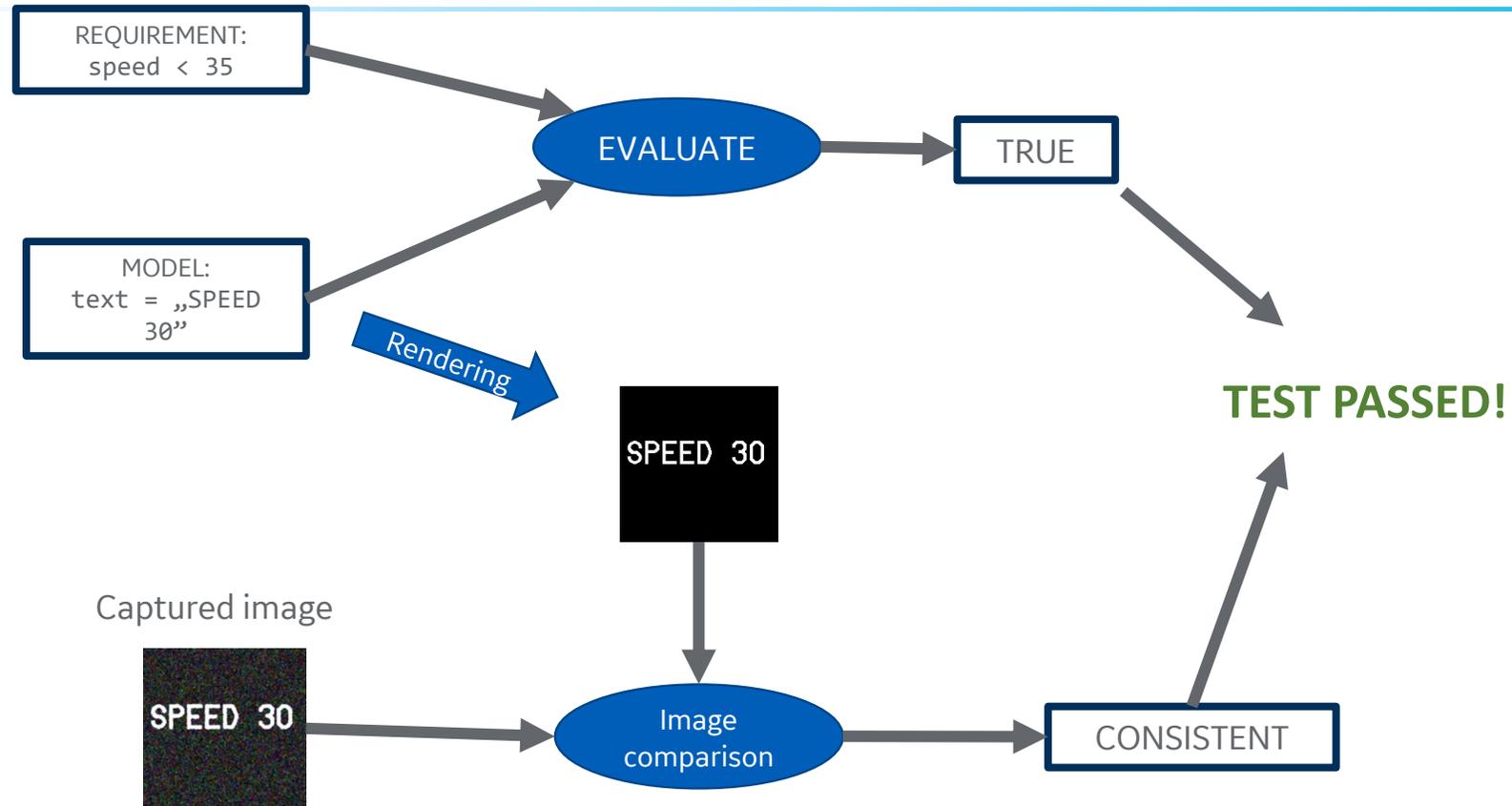
# P-model verification

---

- Consistency between model and captured image is verified by
  - Rendering reference image from model
  - Comparing captured image with reference one
  - If comparison succeeds then the model correctly interprets a captured image
- Requirements are checked against model
  - Substitute all model attributes to formal specification of requirements
  - Check if all assertions evaluate to “TRUTH“



# P-Models: verification

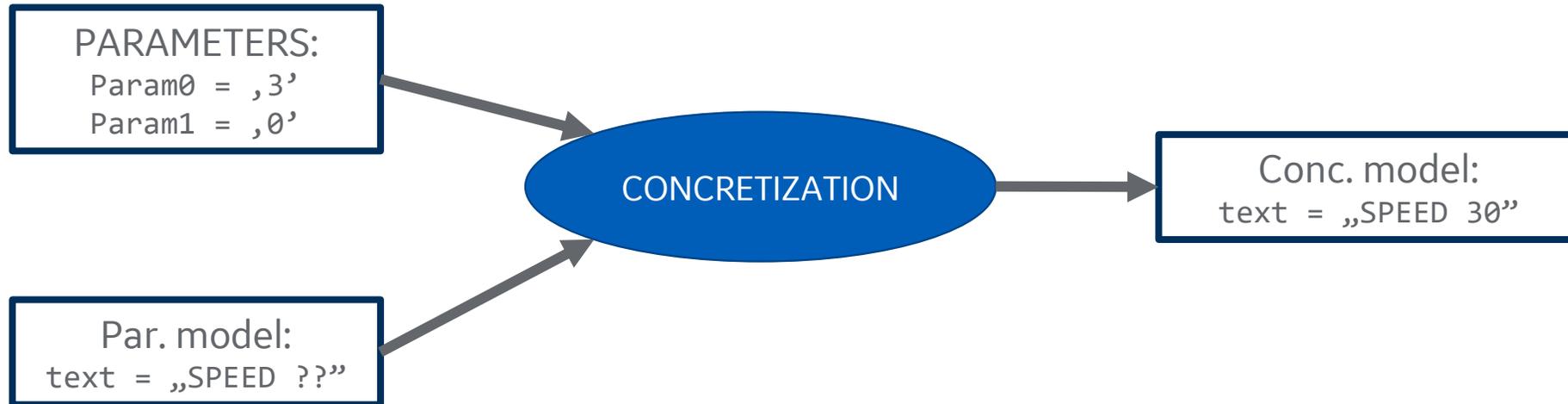


# P-Models - parameters

---

- typed inputs to p-models
- improved flexibility and expresiveness
- parameters are defined at some point.
  - Process known as „concretization”
  - The p-model is then called „concretized model”

# P-Models– concretization



PARAMETRIZED MODEL + PARAMETERS = CONCRETIZED MODEL

# Parameters – How can we find their value?

---

- Can be provided by a user
- Can be enumerated algorithmically
- Can be estimated with machine vision



Project: „An innovative tool for automated software testing for safety critical systems in aviation as a confirmation of the competences of Polish scientific staff of WIA on the international arena”. Founding agreement with NCBIR, Programme: Smart Growth Operational Programme 2014-2020, No: POIR.04.01.04-00-0121/16 signed 04/13/2017



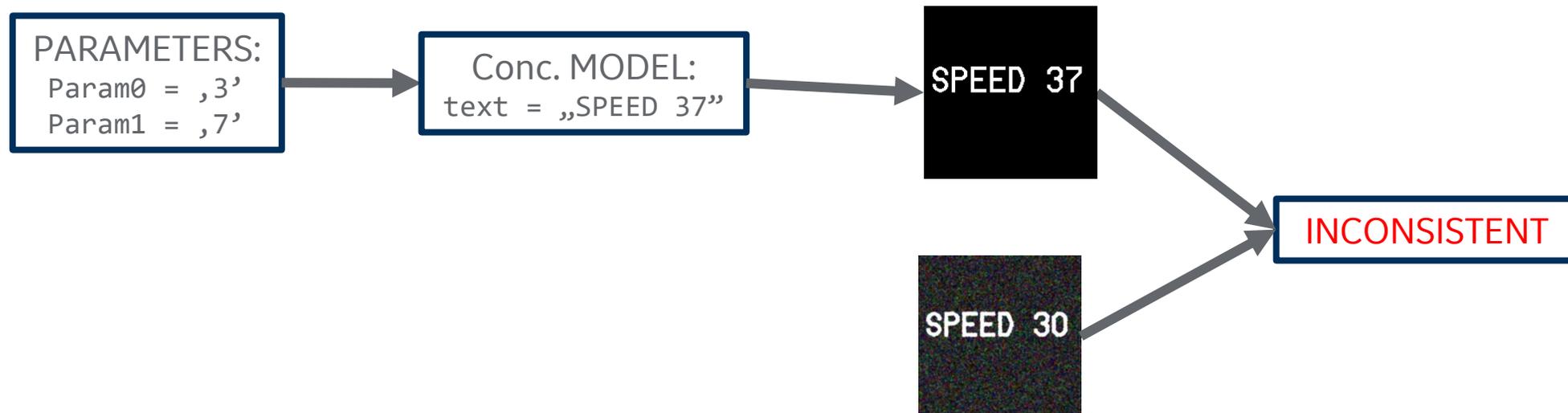
# Parameters – How can we trust their value?

---

- All provided parameters are actually candidate parameters
- Parameters are verified by comparison of captured image with reference image generated from the parameters
- Thus a source of parameters does not matter
- The origin of parameters is put outside of qualification credits

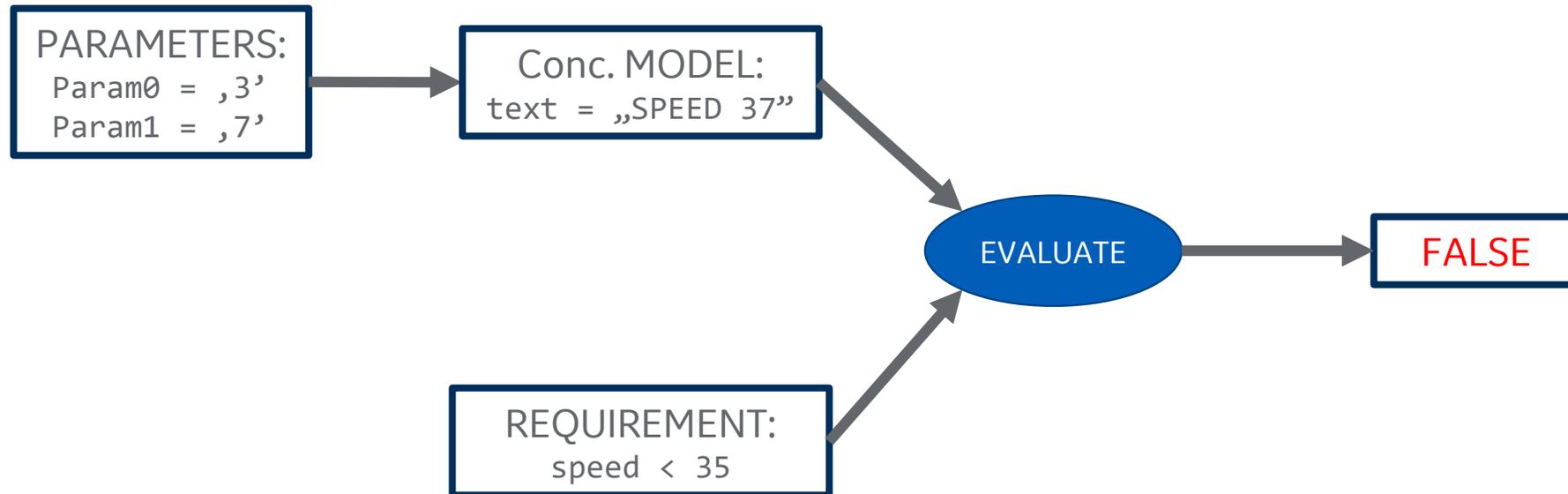
# Parameters – How can we trust their value?

- Invalid parameter will either:
  - Cause failed comparison between reference and capture image



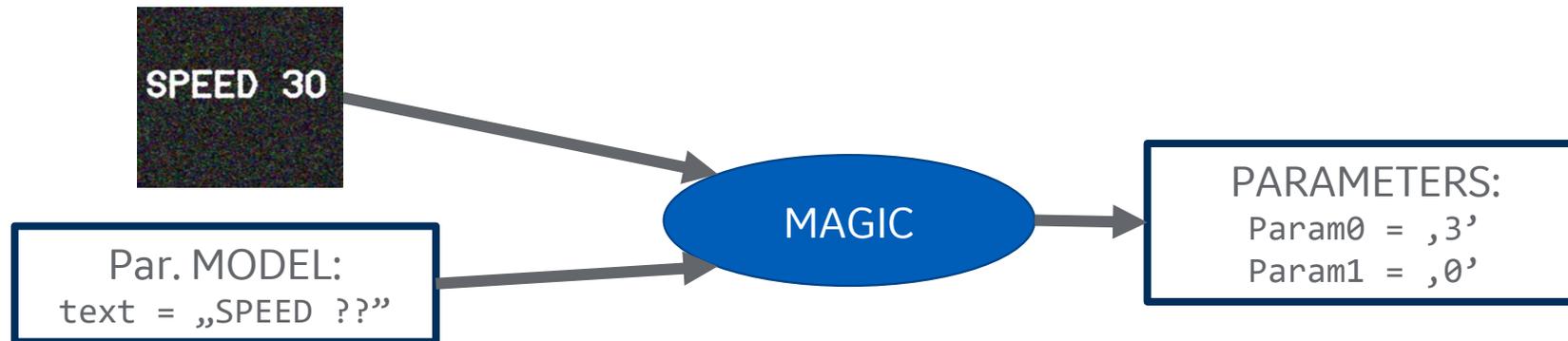
# Parameters – How can we trust their value?

- Or cause failure during testing if requirements are satisfied



# MAGIC

- Magic Algorithms for Graphical Interface Concretization (MAGIC)
- Software library for handling MAGIC models
- Map an image to candidate parameters



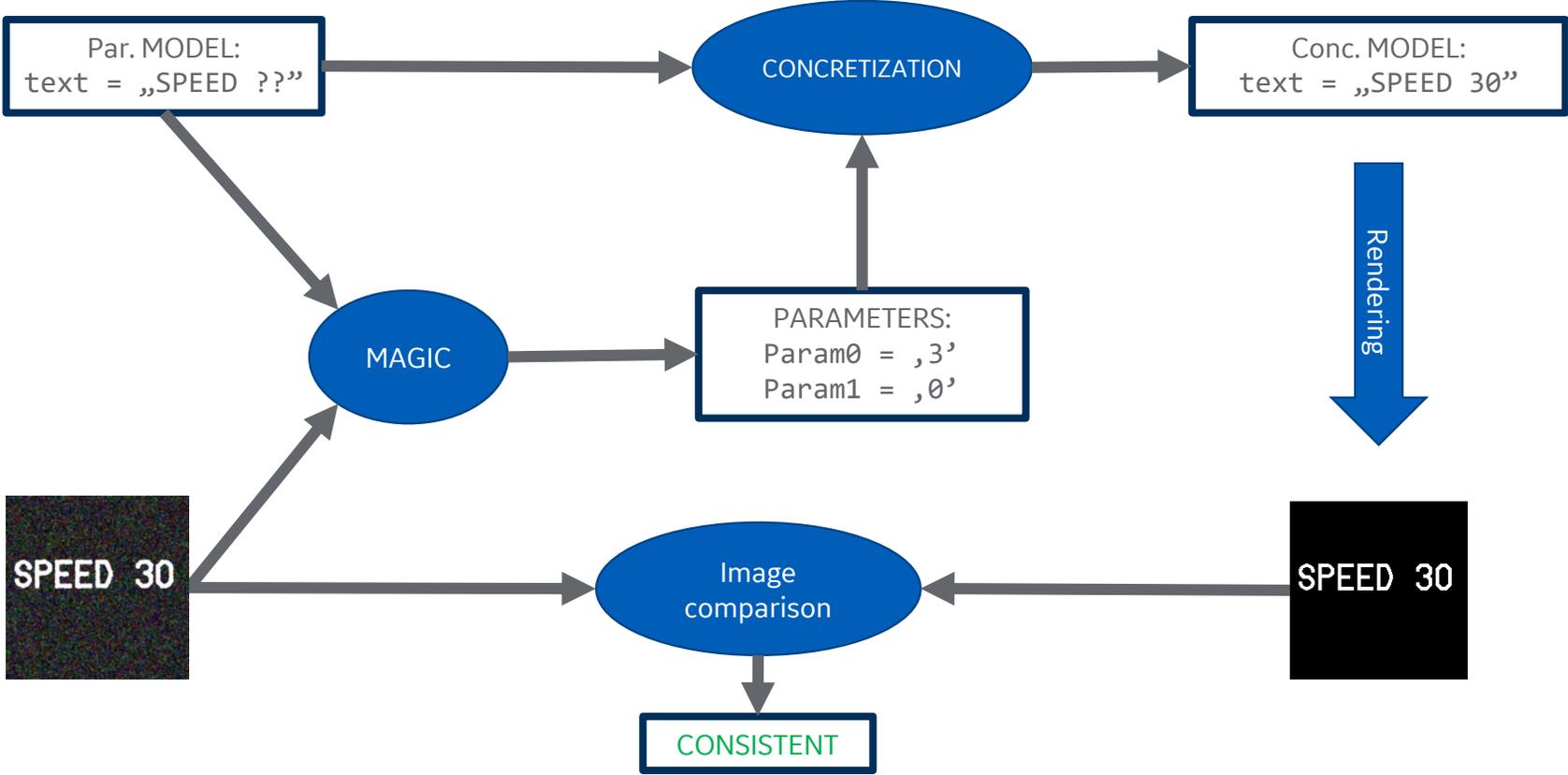
# MAGIC – Algorithms

---

- Discrete optimization  
Find parameters by minimizing difference between rendered from models with captured image  
Effective for small domains and small number of parameters  
Effective if parameters have independent impact on rendered image
- Machine Learning (ML)  
Deep Neural Network that maps image to parameters
- User defined  
MAGIC provides interface for user-defined MAGIC models  
User-defined models can be combined to larger ones



# MAGIC – Verification of candidate parameters



# Test PASS Displays – pros

---

- Great flexibility for testers
  - Models guide machine how to interpret images
  - Models provide structured interpretation of images
- Capability for formal specification for requirements
  - Tester can precisely describe features to be tested rather than saying „image looks correct” in a test procedure
- Allows automated reasoning about displays
  - Eliminates human observer from a test

# Test PASS Displays – pros

---

- Complex machine vision and machine learning algorithms are moved outside qualification credits
- Tester **DOES NOT** have to implement and/or qualify any image recognition algorithms
  - „If one can render it, then one can test it” philosophy
- Expected reductions of costs of testing:
  - Implementation of tests
  - Performing tests

# Test PASS Displays vs Model-based Design

---

- „Correct-by-Construction” principle
- Use model to generate the actual implementation
- Generally models have states and complex logic
- Require comprehensive syntax
- Issues with usage for existing/legacy products
  - Except Test validation/generation
- Actually complementary to Test PASS
  - Test PASS Displays offer means of expressing reqt and tests

# Test PASS Displays vs Machine Vision

---

- Qualification of ML algorithms requires a lot of training data
  - representative
  - labeled
  - expensive
- In Test PASS:
  - algorithms are not qualified
  - each output is verified for each execution of a test
  - false negatives are only expected type of error

# Q&A



