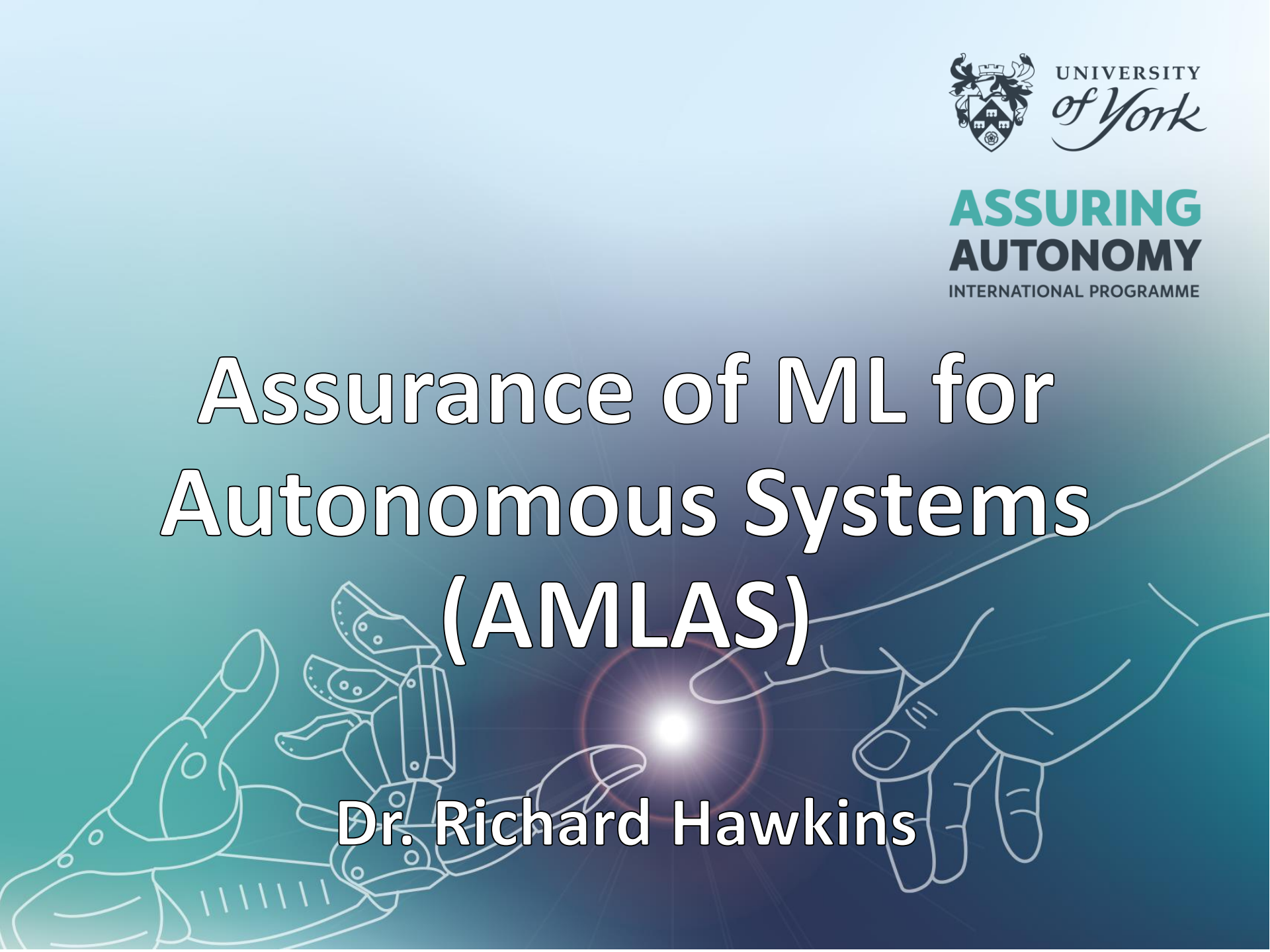
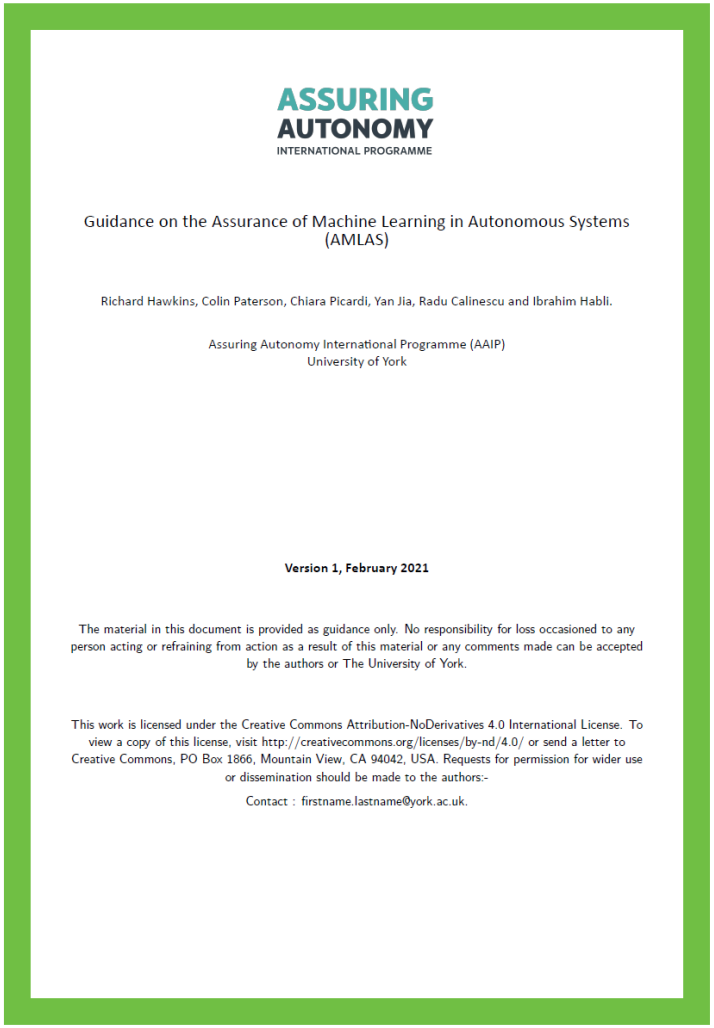

UNIVERSITY of York
ASSURING AUTONOMY
INTERNATIONAL PROGRAMME
Assurance of ML for Autonomous Systems (AMLAS)
Dr. Richard Hawkins

What is AMLAS?



ASSURING AUTONOMY
INTERNATIONAL PROGRAMME

Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS)

Richard Hawkins, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu and Ibrahim Habli.

Assuring Autonomy International Programme (AAIP)
University of York

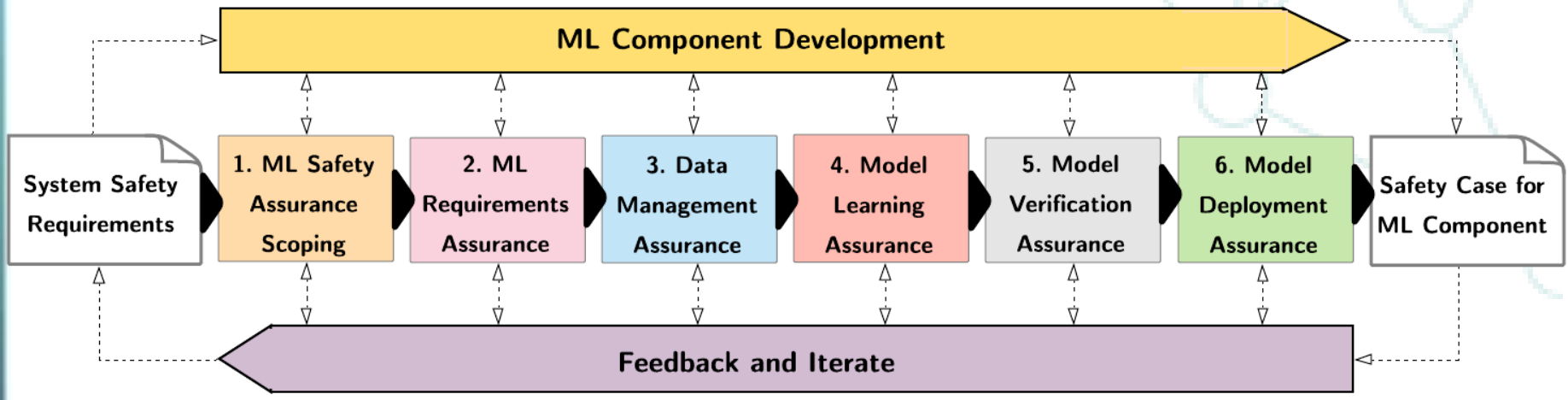
Version 1, February 2021

Contact : firstname.lastname@york.ac.uk.

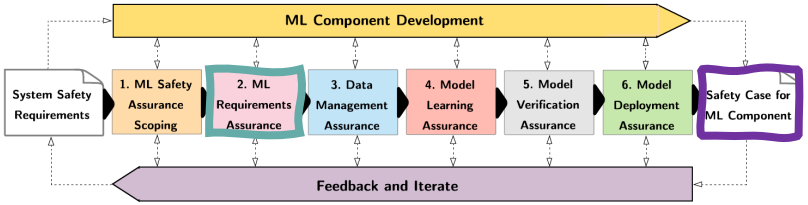
- AMLAS provides
  - Defined **process**
  - Set of **safety case patterns**
- AMLAS enables
  1. Integration of safety assurance into development of ML components
  2. Generation of evidence base for justifying acceptable safety
- Resulting in structured safety case for ML component

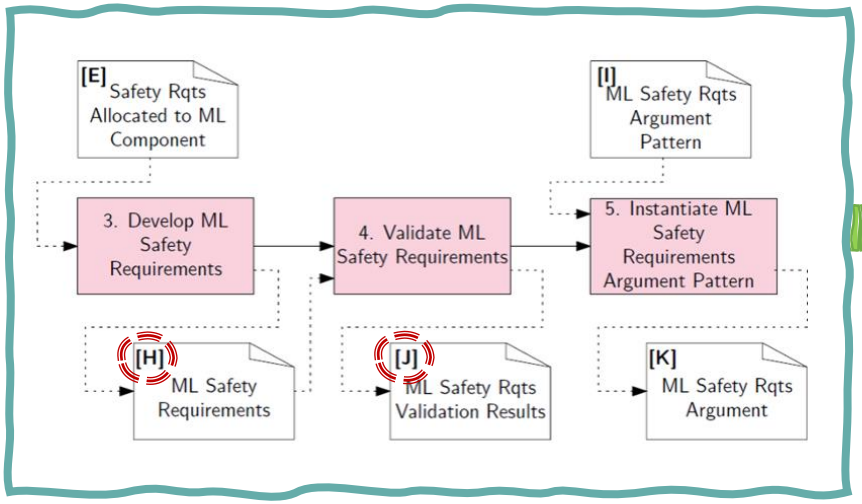
https://www.york.ac.uk/assuring-autonomy/guidance/amlas/

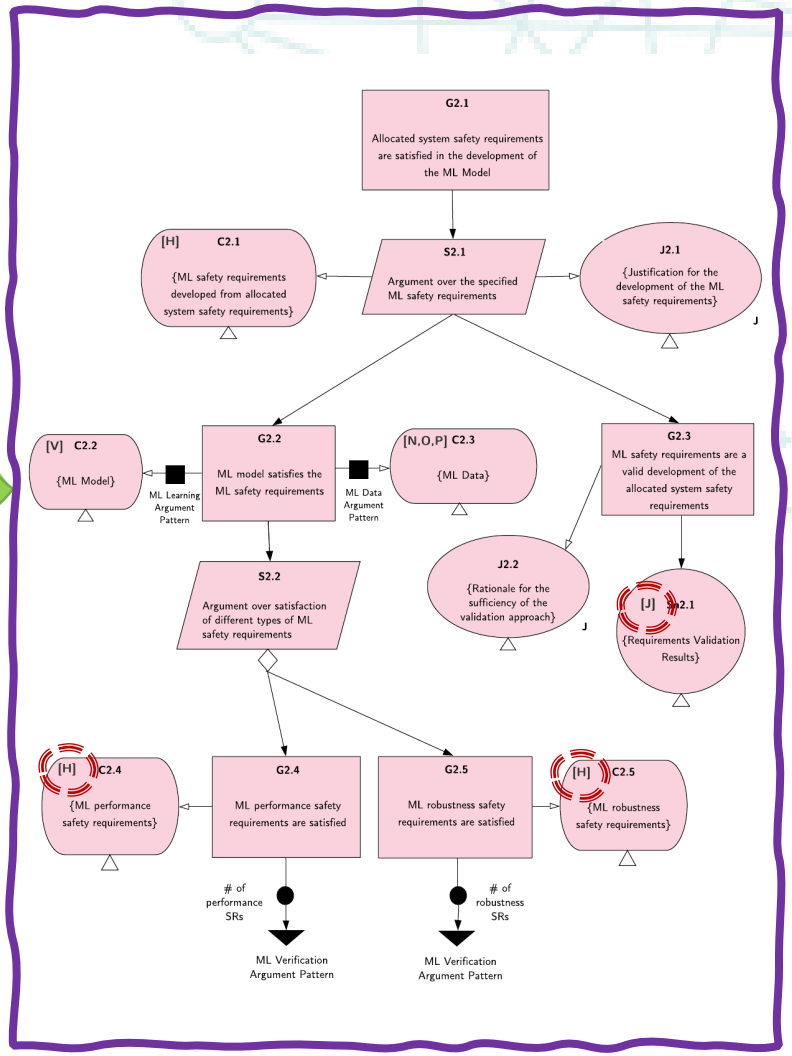
AMLAS Overview



- For each stage we provide
  - Process description
  - Safety argument pattern


ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate


[E] Safety Rqts Allocated to ML Component
[I] ML Safety Rqts Argument Pattern
3. Develop ML Safety Requirements
4. Validate ML Safety Requirements
5. Instantiate ML Safety Requirements Argument Pattern
[H] ML Safety Requirements
[J] ML Safety Rqts Validation Results
[K] ML Safety Rqts Argument


G2.1
Allocated system safety requirements are satisfied in the development of the ML Model
[H] C2.1
{ML safety requirements developed from allocated system safety requirements}
S2.1
Argument over the specified ML safety requirements
J2.1
{Justification for the development of the ML safety requirements}
[V] C2.2
{ML Model}
ML Learning Argument Pattern
G2.2
ML model satisfies the ML safety requirements
ML Data Argument Pattern
[N,O,P] C2.3
{ML Data}
G2.3
ML safety requirements are a valid development of the allocated system safety requirements
S2.2
Argument over satisfaction of different types of ML safety requirements
J2.2
{Rationale for the sufficiency of the validation approach}
[J] J2.1
{Requirements Validation Results}
[H] C2.4
{ML performance safety requirements}
G2.4
ML performance safety requirements are satisfied
G2.5
ML robustness safety requirements are satisfied
[H] C2.5
{ML robustness safety requirements}
# of performance SRs
# of robustness SRs
ML Verification Argument Pattern
ML Verification Argument Pattern

Guidance Structure

Stage 1. ML Safety Assurance Scoping

Objectives

1. Define the scope of the safety assurance process for the ML component.
2. Define the scope of the safety case for the ML component.
3. Create the top-level safety assurance claim and specify the relevant contextual information for the ML safety argument.

Inputs to the Stage

[A] : System Safety Requirements

[B] : Description of Operating Environment of System

[C] : System Description

[D] : ML Component Description

[F] : ML Assurance Scoping Argument Pattern

Outputs of the Stage

[E] : Safety Requirements Allocated to ML Component

[G] : ML Safety Assurance Scoping Argument

Description of the Stage

As shown in Figure 2[1], this stage consists of two activities that are performed to define the safety assurance scope for an ML component. The artefacts generated from this stage are used to instantiate the ML safety assurance scoping argument pattern as part of Activity 2. An ML component comprises an ML model, e.g. a neural network, that is deployed onto the intended computing platform (i.e. comprising both hardware and software).

Additional guidance on the use of ML for autonomous systems can be found at [9].

Activity 1: Define the Safety Assurance Scope for the ML Component [E]

This activity requires as input the system safety requirements ([A]), descriptions of the system and the operating environment ([B], [C]), and a description of the ML component that is being considered ([D]). These inputs shall be used to determine the safety requirements that are allocated to the ML component.

The safety requirements allocated to the ML component shall be defined to control the risk of the identified contributions of the ML component to system hazards. This shall take account of the defined system architecture and the operating environment. At this stage the requirement is independent of any ML technology or metric but instead reflects the need for the component to perform safely with the system regardless of the technology later deployed. The safety requirements allocated to the ML component generated from this activity shall be explicitly documented ([E]).

[1] In the AMLAS process diagrams, rectangles represent activities. Document symbols represent input or output artefacts. Each document symbol has a unique ID (top left) that is used to refer to the artefact in the guidance text or the argument pattern, e.g. [A] is a reference to artefact A.

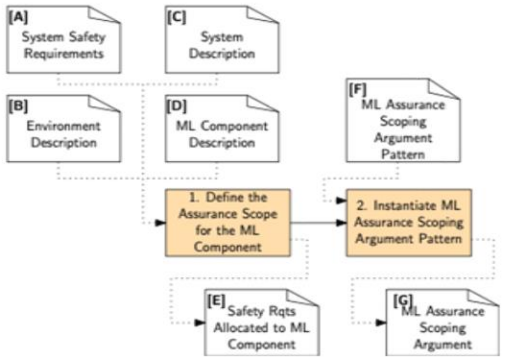


Figure 2: AMLAS ML Assurance Scoping Process

Example 1.

Consider an autonomous driving application in which a subsystem may be required to identify pedestrians at a crossing. A component within the perception pipeline may have a requirement of the form "When Ego is 50 metres from the crossing, the object detection component shall identify pedestrians that are on or close to the crossing in their correct position."

Note 1.

The allocation of safety requirements must consider architectural features such as redundancy when allocating the safety requirements to the ML component. Where redundancy is provided by other, non-machine-learnt components, this may reduce the assurance burden on the ML component that should be reflected in the allocated safety requirements.

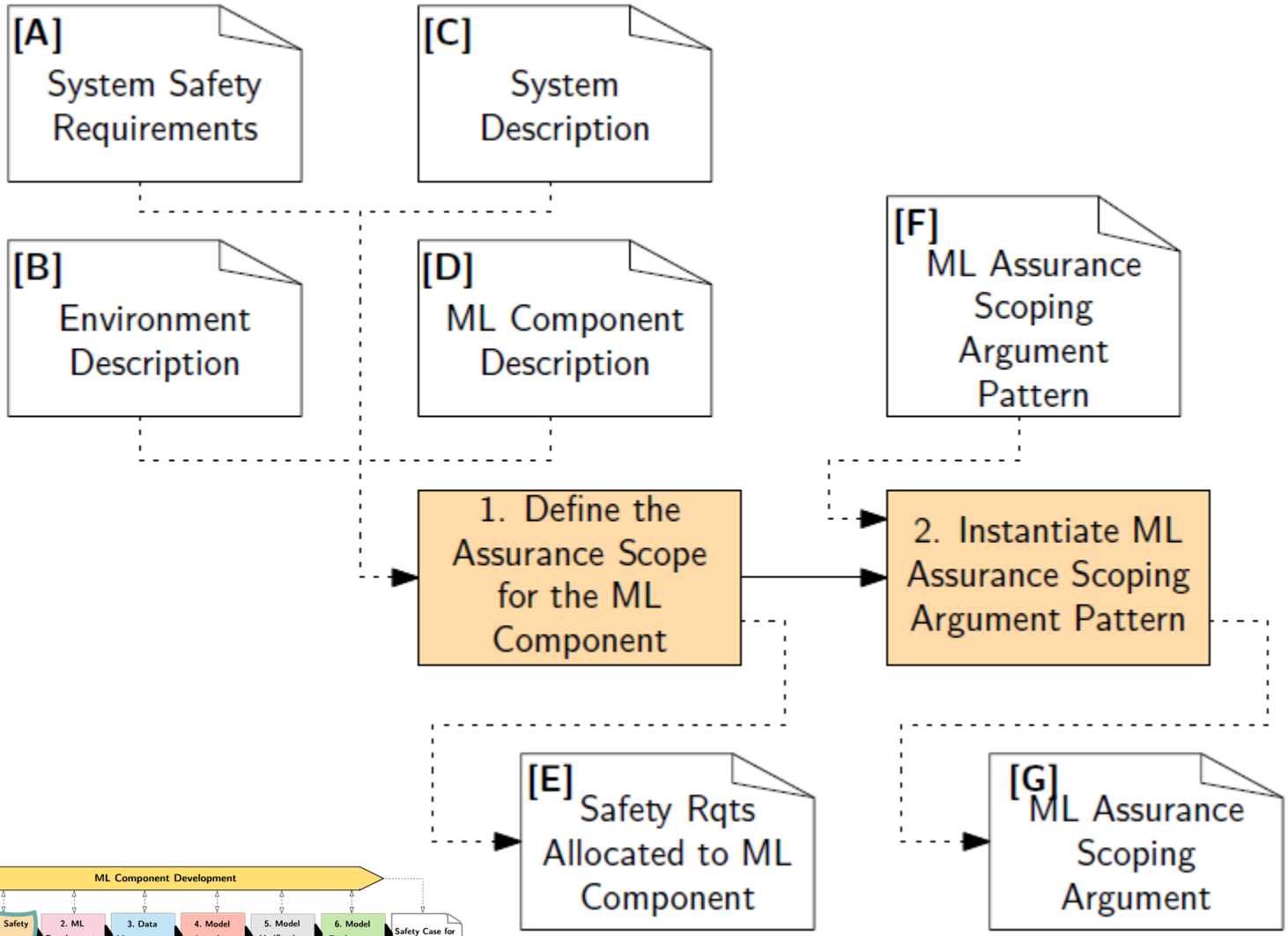
Note 2.

The contribution of the human as part of the broader system should be considered. A human may provide, for example, oversight or fallback in the case of failure of the ML component. These human contributions, and any associated human factors issues, e.g. automation bias [59], should be reflected when allocating safety requirements to the ML component.

Artefact [A]: System Safety Requirements

The safety requirements are generated from the system safety assessment process. Such a process covers hazard identification and risk analysis. Importantly, it shall determine the contribution, i.e. in the form of concrete failure conditions, that the output of the machine learning component makes to potential system hazards. A simplified linear chain of events that links a machine learning failure with a hazard is illustrated in Figure 3.

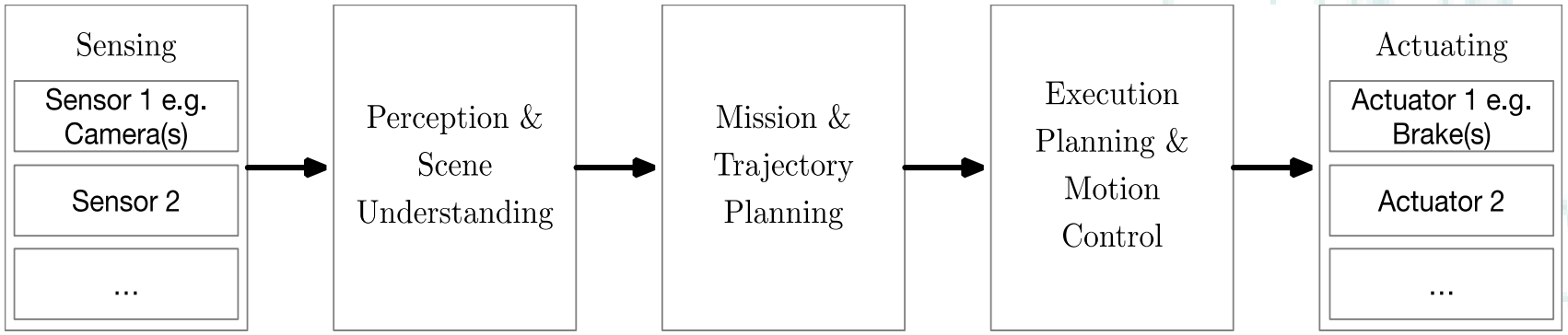
1. ML Safety Assurance Scoping

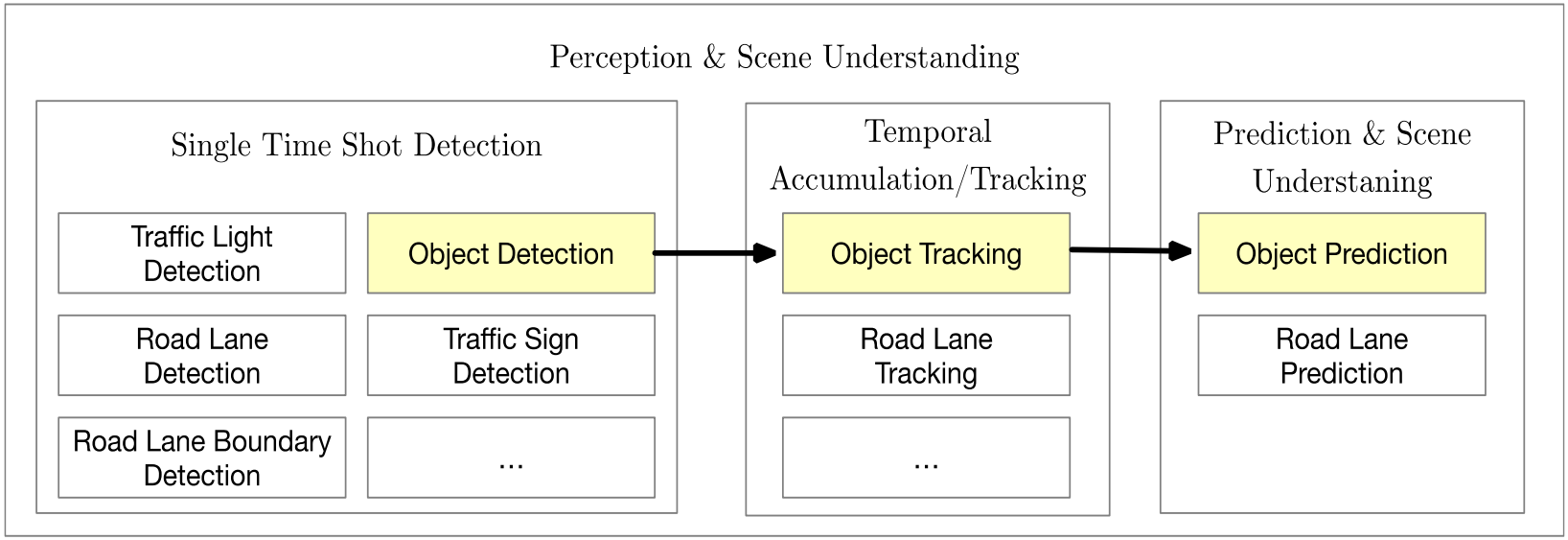


Scoping Example

- Understand and define
  - **System Architecture (see next slide)**
  - **System Operating Environment (see below)**
  - **System Safety Requirements**

Model Integration

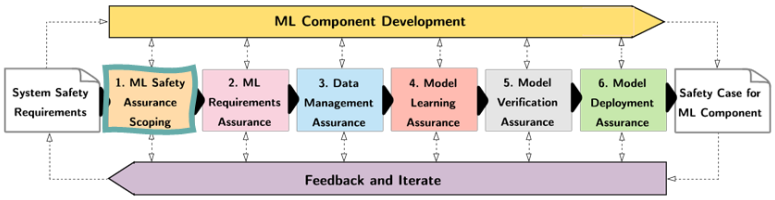


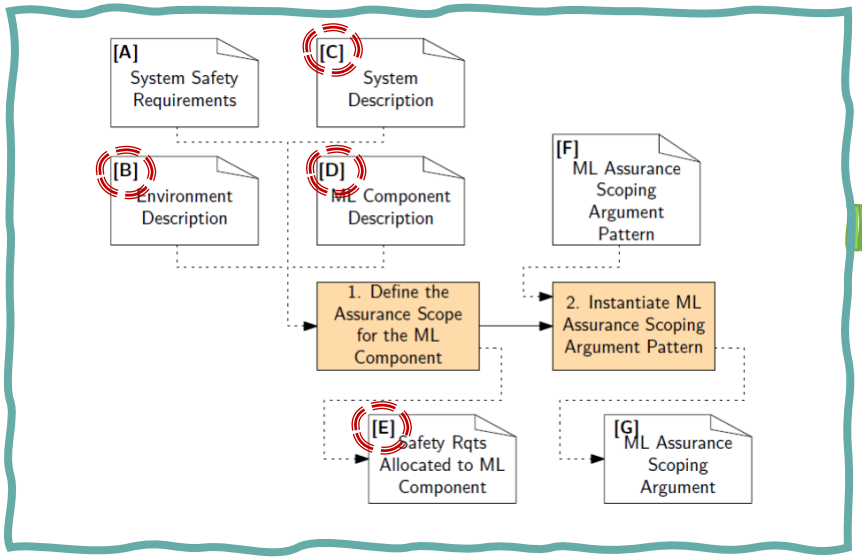


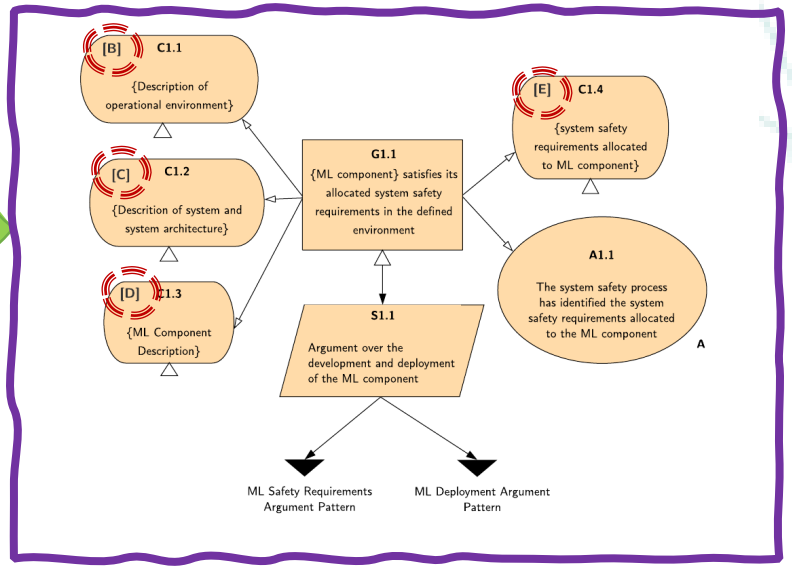
Allocating safety requirements

Example 1.

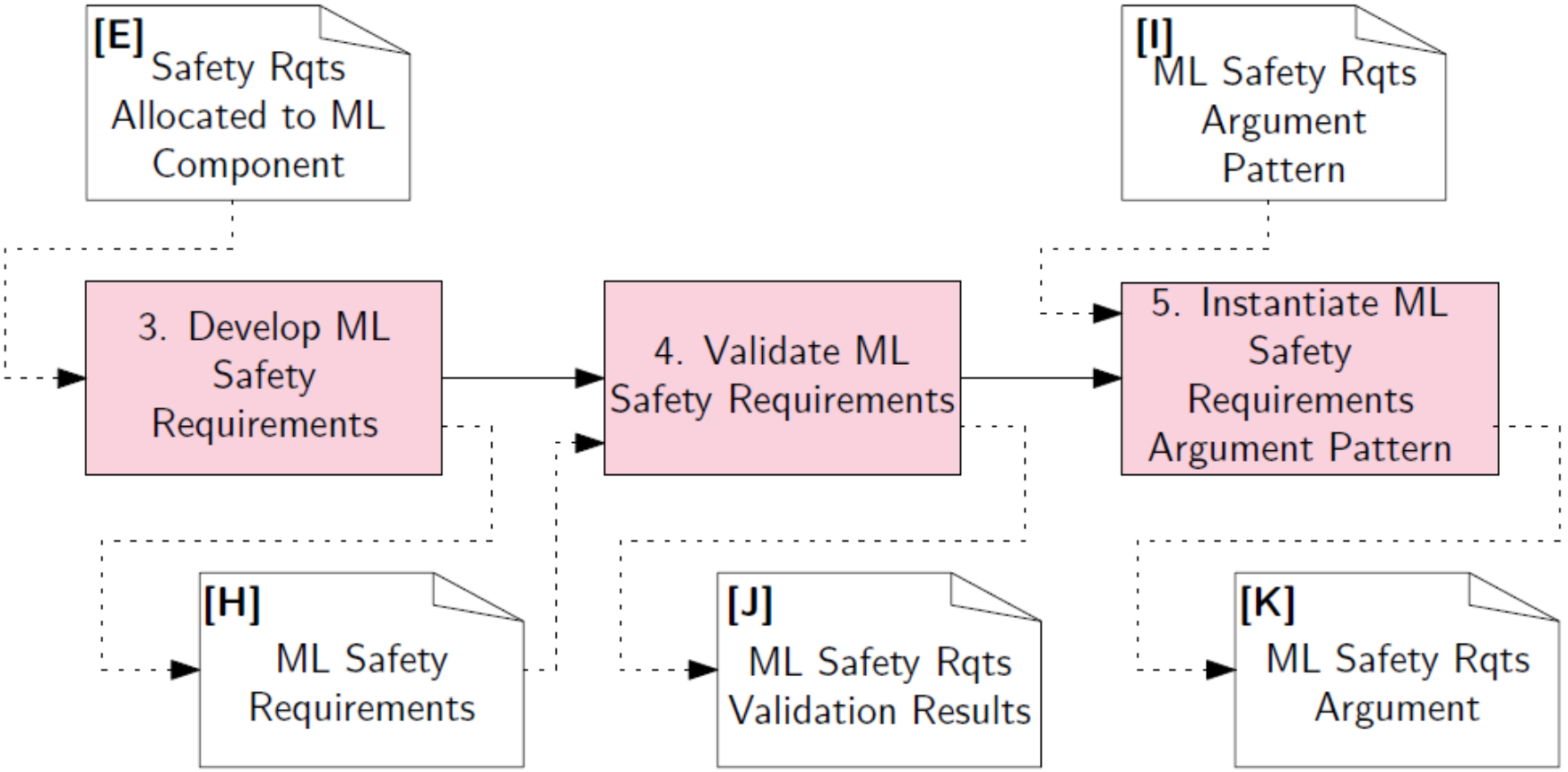
Consider an autonomous driving application in which a subsystem may be required to identify pedestrians at a crossing. A component within the perception pipeline may have a requirement of the form "When Ego is 50 metres from the crossing, the object detection component shall identify pedestrians that are on or close to the crossing in their correct position."

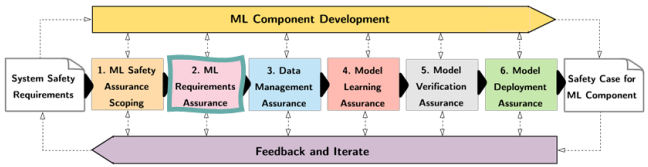

ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate


[A] System Safety Requirements
[C] System Description
[B] Environment Description
[D] ML Component Description
[F] ML Assurance Scoping Argument Pattern
1. Define the Assurance Scope for the ML Component
2. Instantiate ML Assurance Scoping Argument Pattern
[E] Safety Rqts Allocated to ML Component
[G] ML Assurance Scoping Argument


[B] C1.1
{Description of operational environment}
[C] C1.2
{Descrition of system and system architecture}
[D] C1.3
{ML Component Description}
G1.1
{ML component} satisfies its allocated system safety requirements in the defined environment
[E] C1.4
{system safety requirements allocated to ML component}
A1.1
The system safety process has identified the system safety requirements allocated to the ML component
S1.1
Argument over the development and deployment of the ML component
ML Safety Requirements Argument Pattern
ML Deployment Argument Pattern

2. ML Safety Rqts Assurance





Environment Context

What are the key features of the operating environment?

- **People, Vehicles, weather conditions, lighting, road type, etc., etc.**

ML Safety Requirements

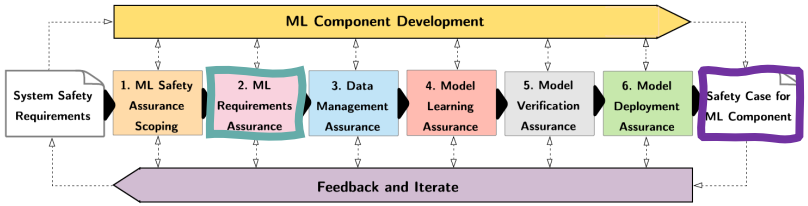
Example 7.

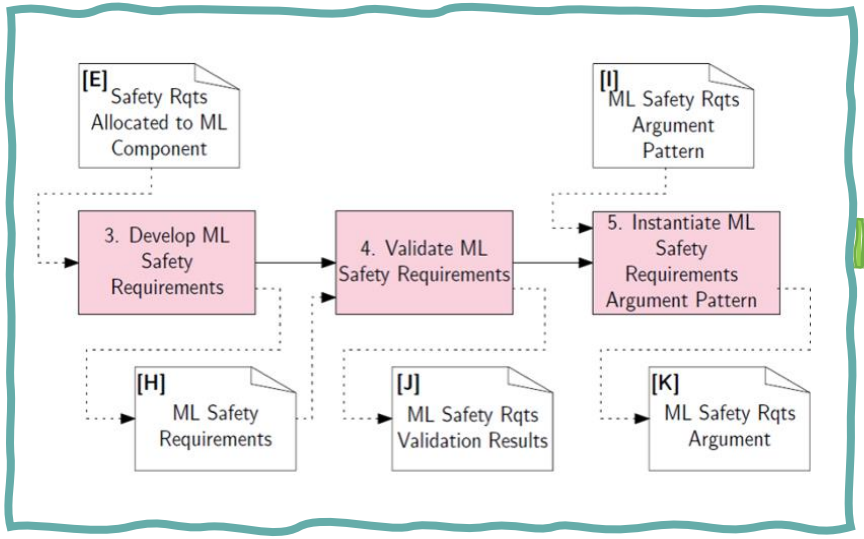
The ML safety requirement presented in Example 1 may now be refined into performance and robustness requirements [22]. Example performance requirements may include:

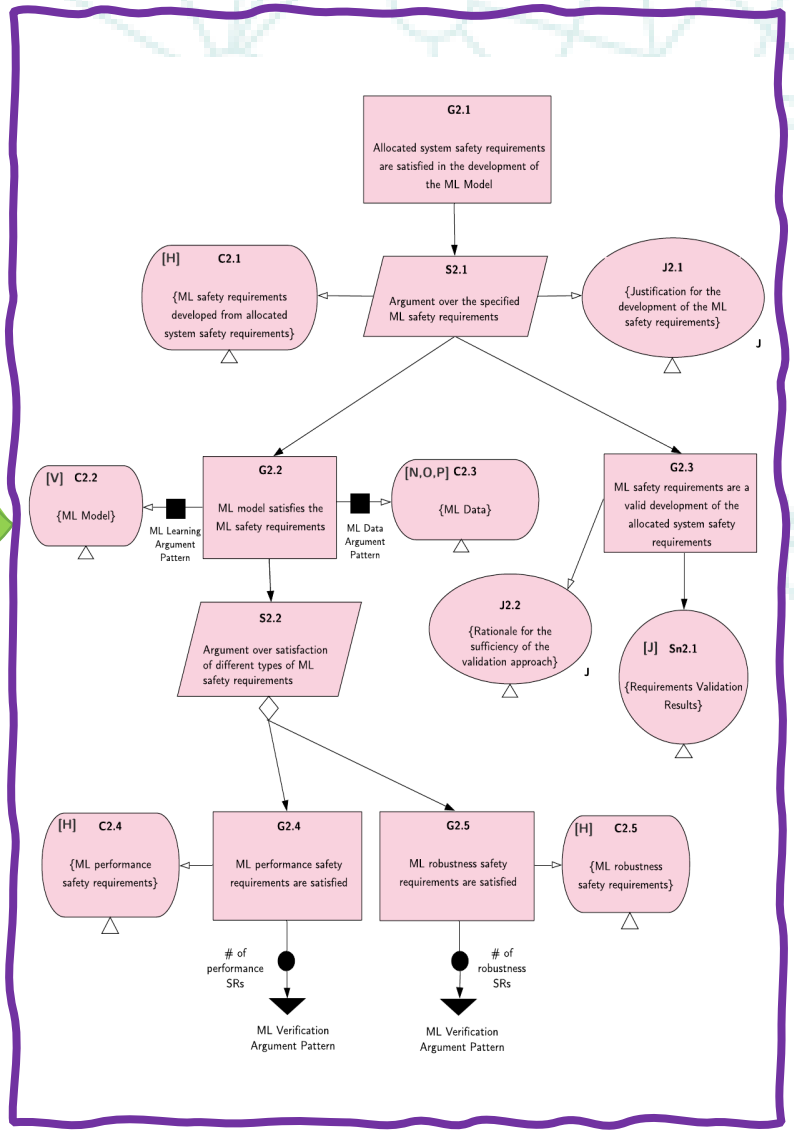
- The ML component shall determine the position of the specified feature in each input frame within 5 pixels of actual position.
- The ML component shall identify the presence of any person present in the defined area with an accuracy of at least 0.93

Example robustness requirements may include:

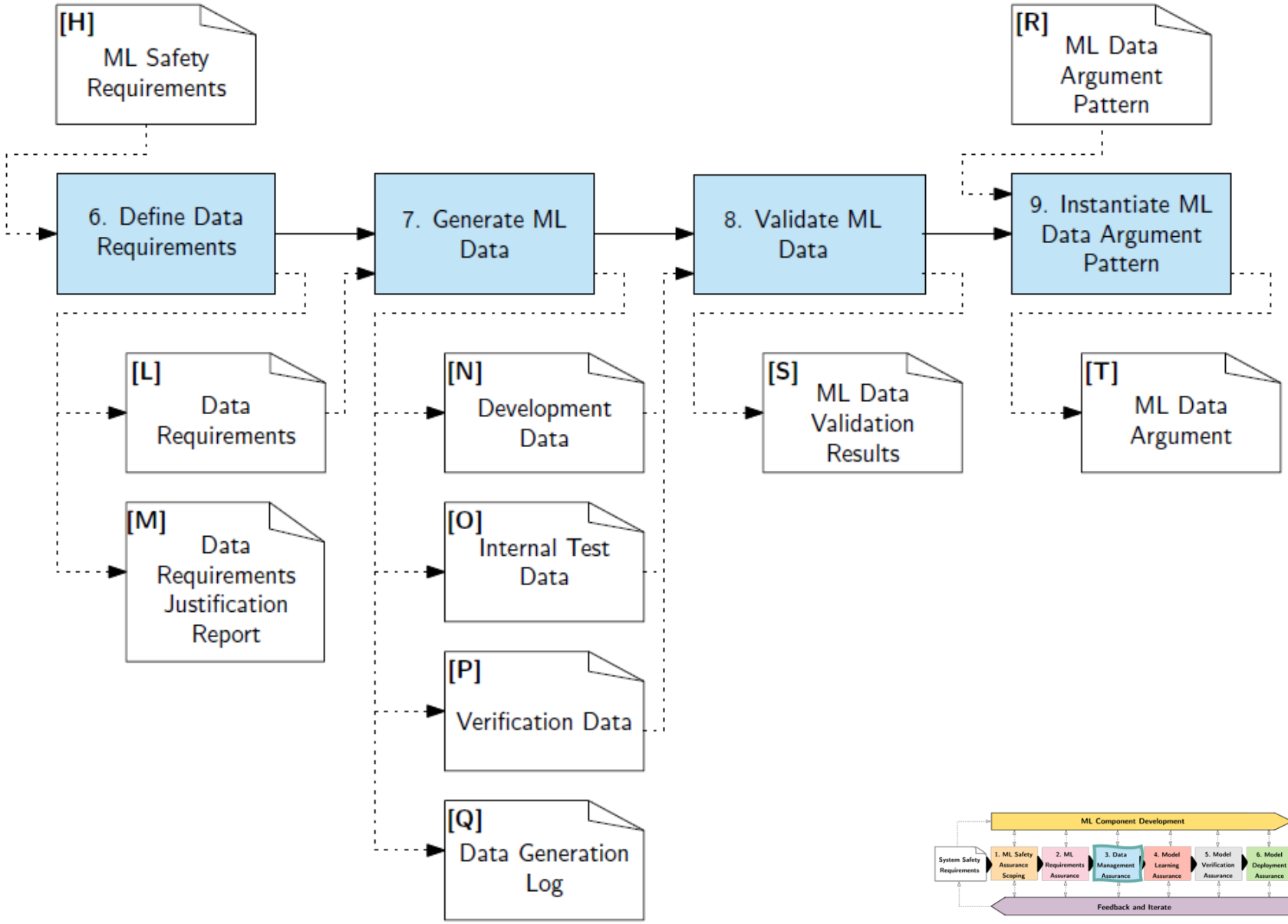
- The ML component shall perform as required in the defined range of lighting conditions experienced during operation of the system.
- The ML component shall identify a person irrespective of their pose with respect to the camera.


ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate


[E] Safety Rqts Allocated to ML Component
[I] ML Safety Rqts Argument Pattern
3. Develop ML Safety Requirements
4. Validate ML Safety Requirements
5. Instantiate ML Safety Requirements Argument Pattern
[H] ML Safety Requirements
[J] ML Safety Rqts Validation Results
[K] ML Safety Rqts Argument


G2.1
Allocated system safety requirements are satisfied in the development of the ML Model
[H] C2.1
{ML safety requirements developed from allocated system safety requirements}
S2.1
Argument over the specified ML safety requirements
J2.1
{Justification for the development of the ML safety requirements}
J
[V] C2.2
{ML Model}
ML Learning Argument Pattern
G2.2
ML model satisfies the ML safety requirements
ML Data Argument Pattern
[N,O,P] C2.3
{ML Data}
G2.3
ML safety requirements are a valid development of the allocated system safety requirements
S2.2
Argument over satisfaction of different types of ML safety requirements
J2.2
{Rationale for the sufficiency of the validation approach}
J
[J] Sn2.1
{Requirements Validation Results}
[H] C2.4
{ML performance safety requirements}
G2.4
ML performance safety requirements are satisfied
G2.5
ML robustness safety requirements are satisfied
[H] C2.5
{ML robustness safety requirements}
# of performance SRs
# of robustness SRs
ML Verification Argument Pattern
ML Verification Argument Pattern

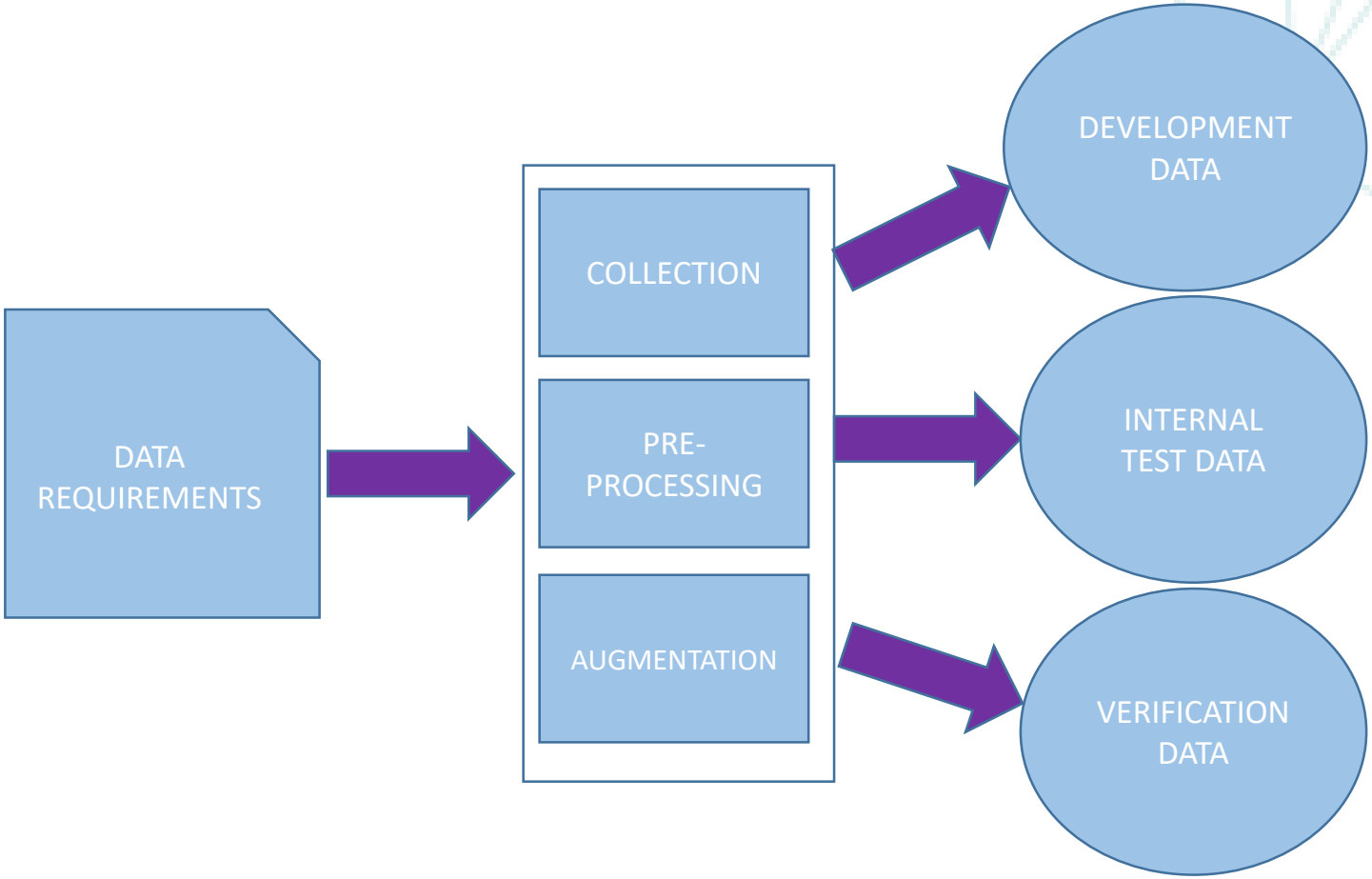
3. Data Management

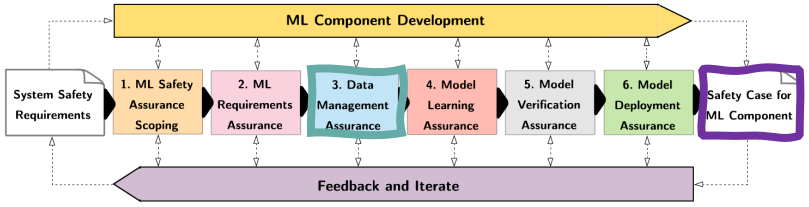


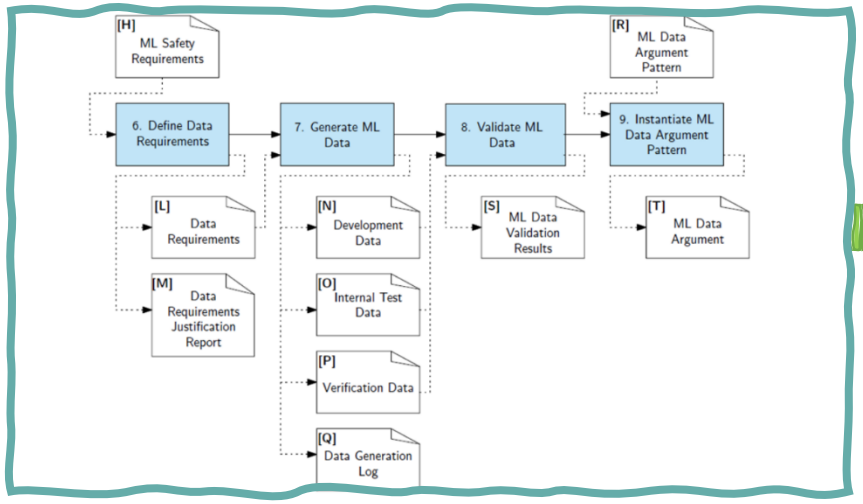
Data Requirements

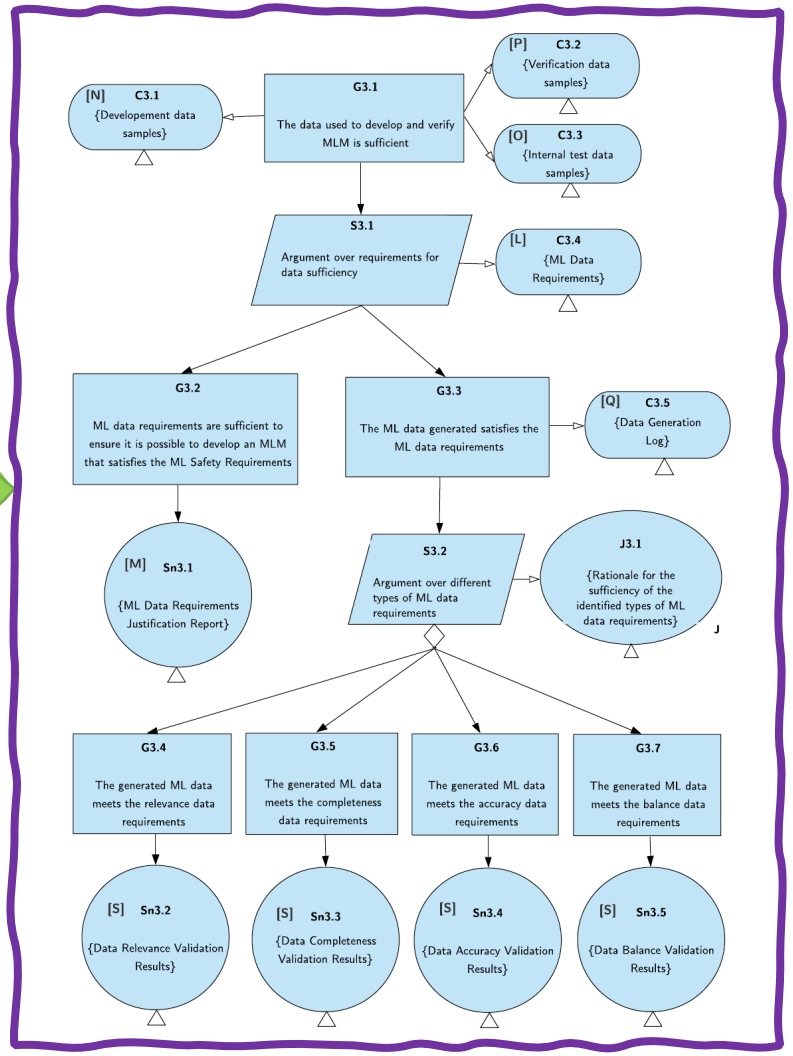
- ML data requirements shall include consideration of:
  - **Relevance**
  - **Completeness**
  - **Accuracy**
  - **Balance**

Generate ML data

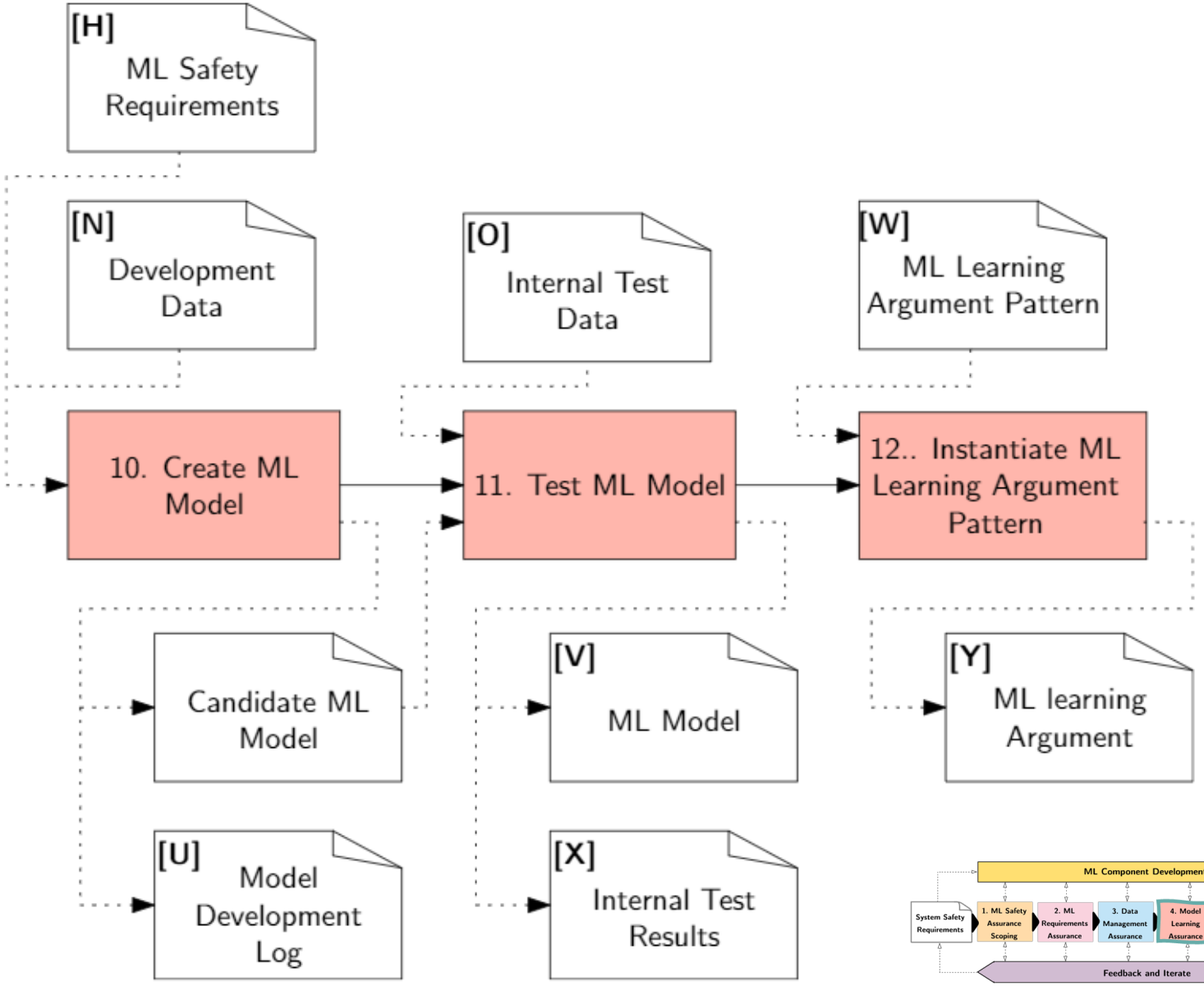



ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate


[H] ML Safety Requirements
[R] ML Data Argument Pattern
6. Define Data Requirements
7. Generate ML Data
8. Validate ML Data
9. Instantiate ML Data Argument Pattern
[L] Data Requirements
[N] Development Data
[S] ML Data Validation Results
[T] ML Data Argument
[M] Data Requirements Justification Report
[O] Internal Test Data
[P] Verification Data
[Q] Data Generation Log


[N] C3.1 {Developement data samples}
G3.1 The data used to develop and verify MLM is sufficient
[P] C3.2 {Verification data samples}
[O] C3.3 {Internal test data samples}
S3.1 Argument over requirements for data sufficiency
[L] C3.4 {ML Data Requirements}
G3.2 ML data requirements are sufficient to ensure it is possible to develop an MLM that satisfies the ML Safety Requirements
G3.3 The ML data generated satisfies the ML data requirements
[Q] C3.5 {Data Generation Log}
[M] Sn3.1 {ML Data Requirements Justification Report}
S3.2 Argument over different types of ML data requirements
J3.1 {Rationale for the sufficiency of the identified types of ML data requirements}
G3.4 The generated ML data meets the relevance data requirements
G3.5 The generated ML data meets the completeness data requirements
G3.6 The generated ML data meets the accuracy data requirements
G3.7 The generated ML data meets the balance data requirements
[S] Sn3.2 {Data Relevance Validation Results}
[S] Sn3.3 {Data Completeness Validation Results}
[S] Sn3.4 {Data Accuracy Validation Results}
[S] Sn3.5 {Data Balance Validation Results}

4. Model Learning



4. Model Learning Assurance

Key Assurance Artefacts

Model Development Log

Q1. What forms of model were considered and on what basis was the model type selected?

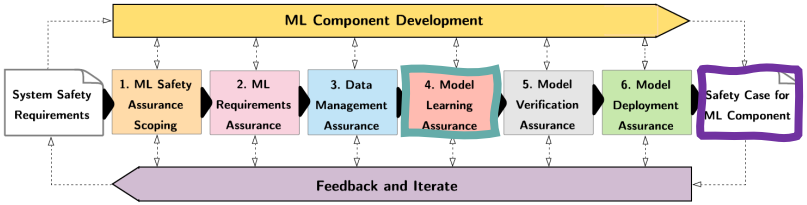
Q2. What approaches were used for hyper parameter tuning e.g. meta learning?

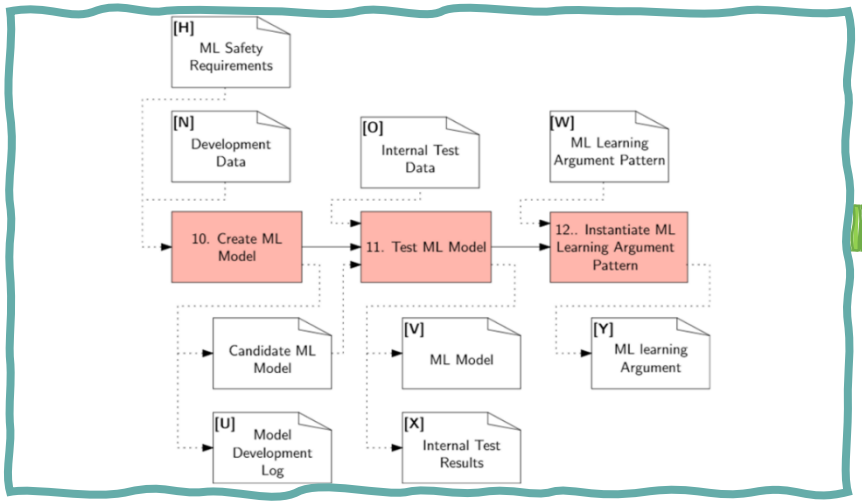
Q3. What techniques were applied to increase the generalizability of the model e.g. drop out?

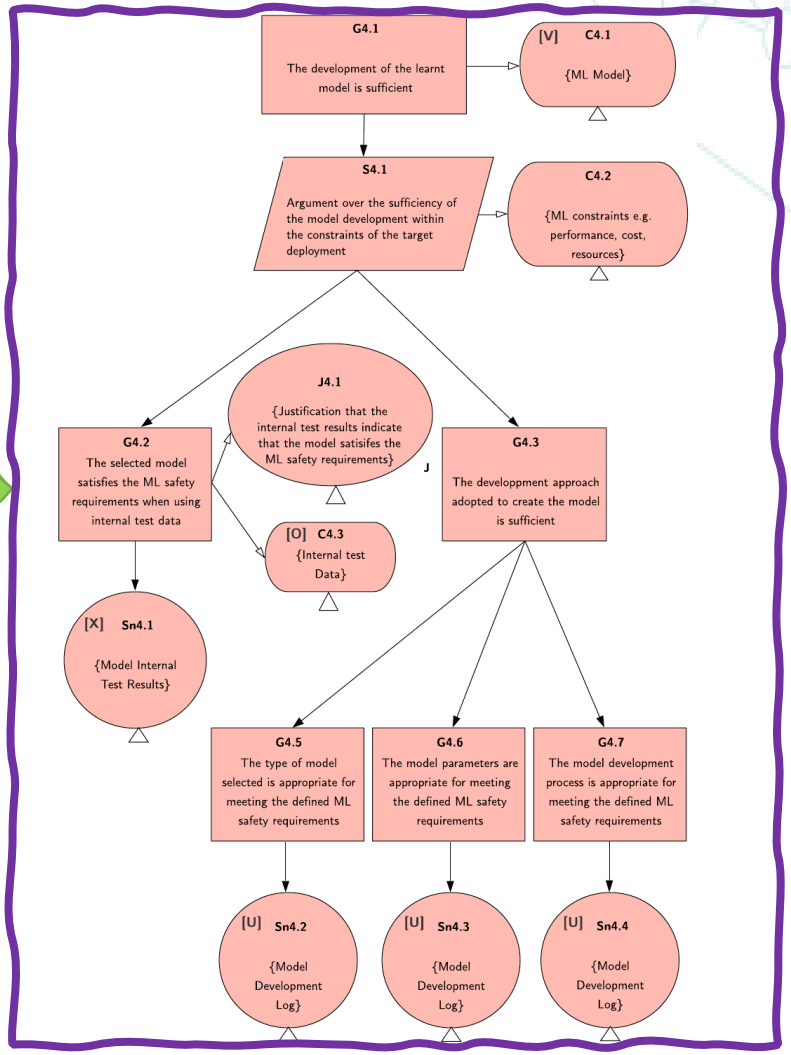
Q4. …

Internal Test Results

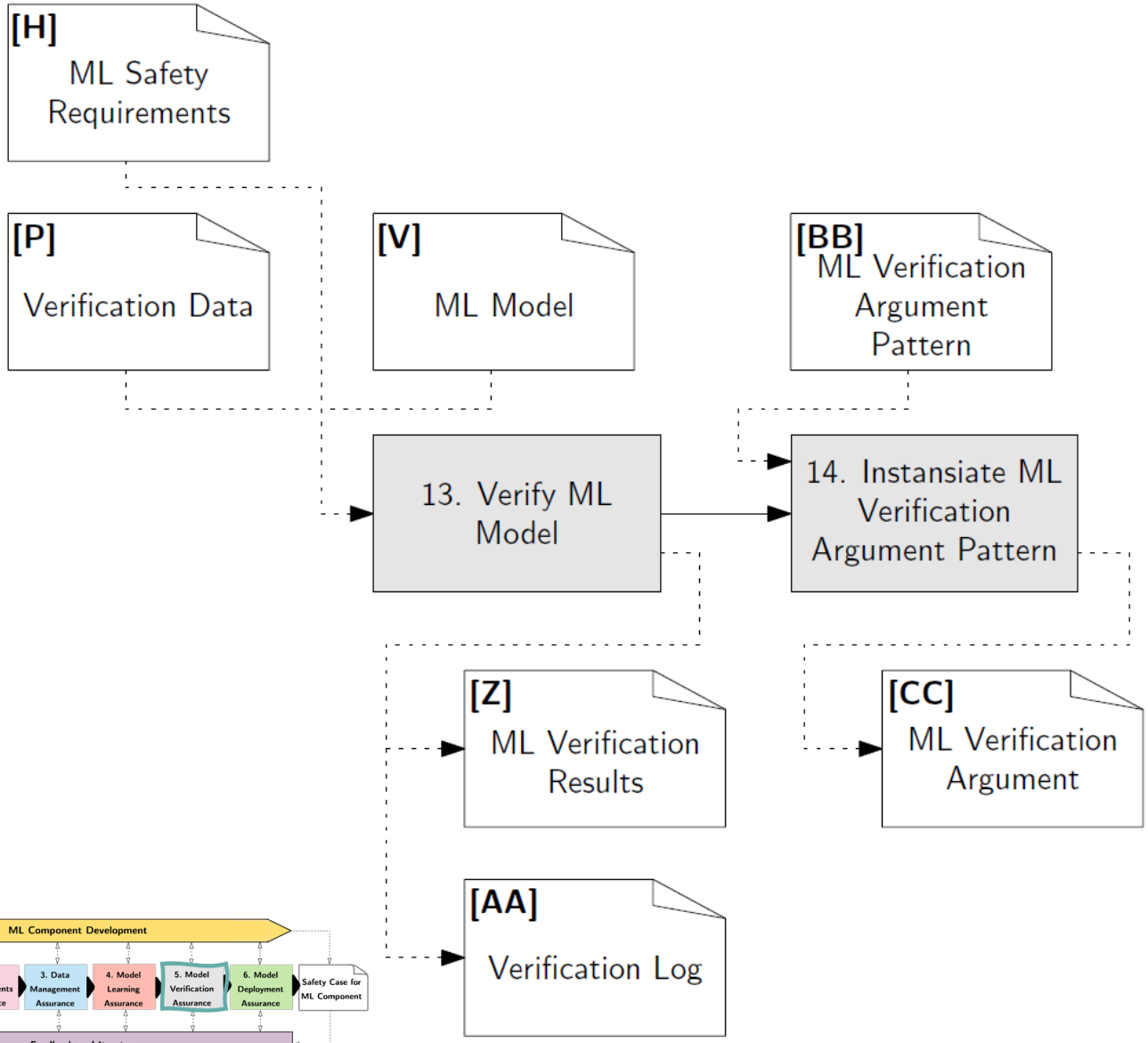
Why was this particular model chosen?


ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate


[H] ML Safety Requirements
[N] Development Data
[O] Internal Test Data
[W] ML Learning Argument Pattern
10. Create ML Model
11. Test ML Model
12.. Instantiate ML Learning Argument Pattern
Candidate ML Model
[V] ML Model
[Y] ML learning Argument
[U] Model Development Log
[X] Internal Test Results


G4.1
The development of the learnt model is sufficient
[V] C4.1
{ML Model}
S4.1
Argument over the sufficiency of the model development within the constraints of the target deployment
C4.2
{ML constraints e.g. performance, cost, resources}
J4.1
{Justification that the internal test results indicate that the model satisifes the ML safety requirements}
G4.2
The selected model satisfies the ML safety requirements when using internal test data
G4.3
The developpment approach adopted to create the model is sufficient
J
[O] C4.3
{Internal test Data}
[X] Sn4.1
{Model Internal Test Results}
G4.5
The type of model selected is appropriate for meeting the defined ML safety requirements
G4.6
The model parameters are appropriate for meeting the defined ML safety requirements
G4.7
The model development process is appropriate for meeting the defined ML safety requirements
[U] Sn4.2
{Model Development Log}
[U] Sn4.3
{Model Development Log}
[U] Sn4.4
{Model Development Log}

5. Model Verification



Test-based Verification

- Use verification data to demonstrate model generalises to cases not present in model learning stage.
- Examine cases on boundaries or which are known to be problematic within the deployment context

Example 34.

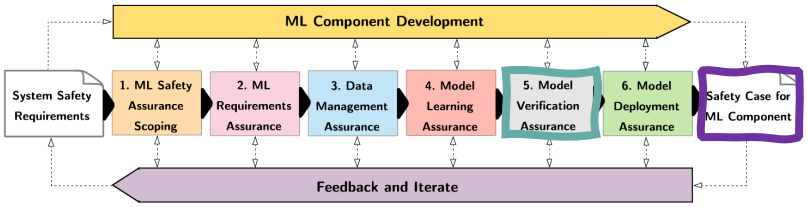
Since we know that material on a camera lens can lead to blurring in regions of an image, we may make use of 'contextual mutators' [52] to assess the robustness of a neural network with respect to levels of blur. In this way the level of blur which can be accommodated can be assessed and related to contextually meaningful measures.

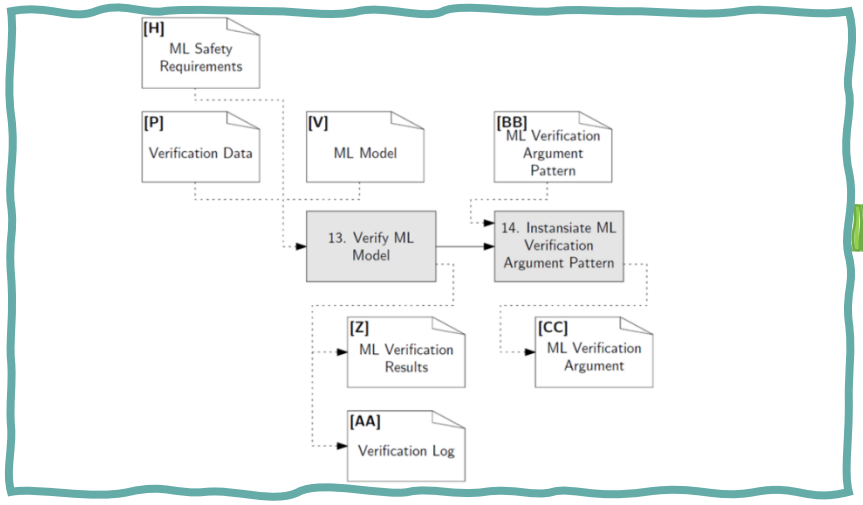
Formal Verification

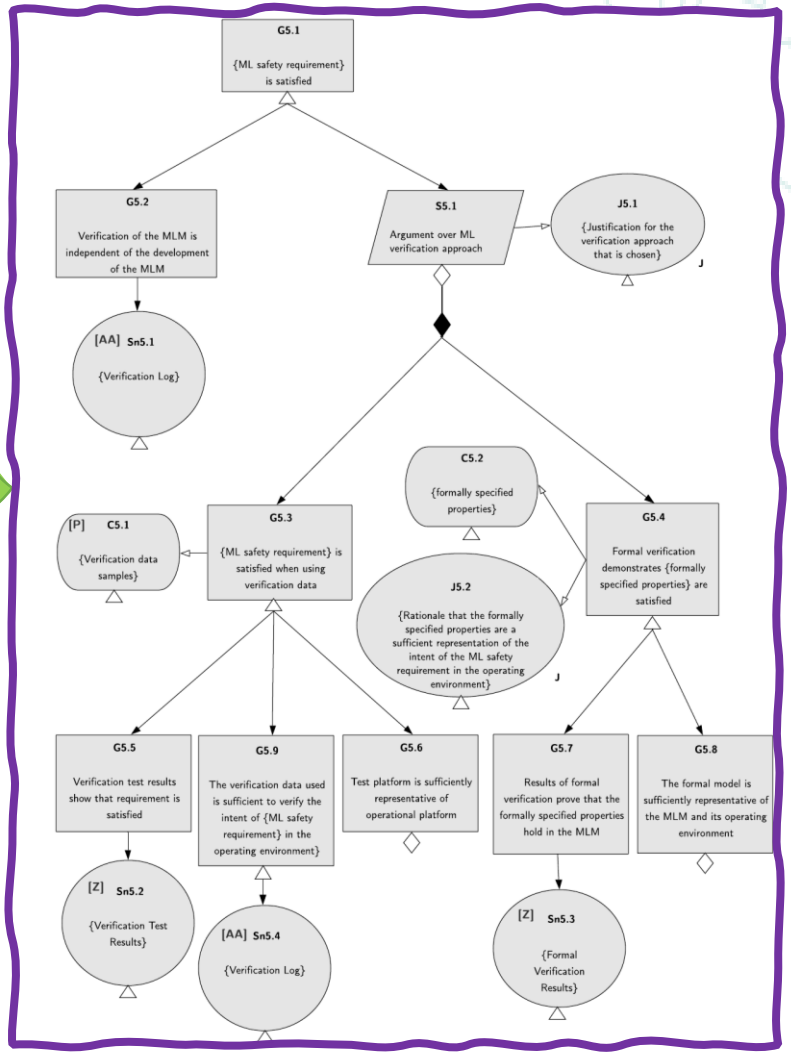
- Use mathematical techniques to prove that the learnt model satisfies formally specified properties
- Formal properties must be derived from the ML safety requirements.
  - Crucial to justify the sufficiency of the translation
  - Formal results must be mapped to operating context

Example 36.

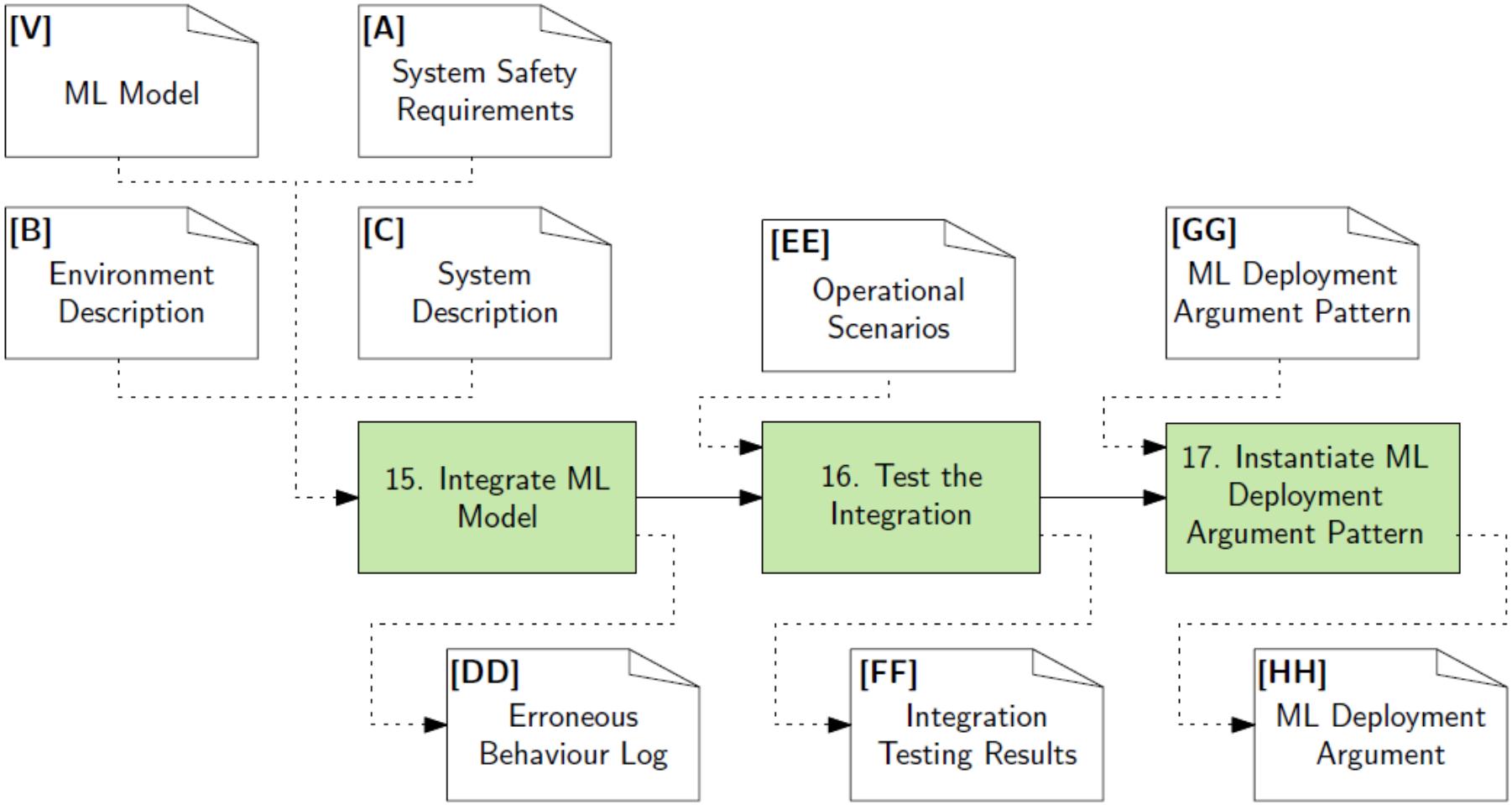
Formal verification of neural networks are able to demonstrate that a perturbation of up to threshold value, ϵ, on all the inputs leads to a classification that is the same as the original sample class [36, 35]. Such a result is only meaningful when the value of ϵ is translated into contextually meaningful values such as steering angles. It is of little value when it simply provides a range of variation in pixel values for a single input sample (point wise verification).

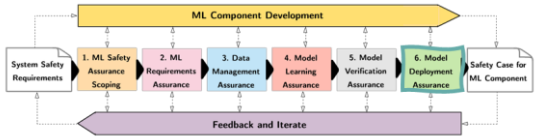

ML Component Development
System Safety Requirements
1. ML Safety Assurance Scoping
2. ML Requirements Assurance
3. Data Management Assurance
4. Model Learning Assurance
5. Model Verification Assurance
6. Model Deployment Assurance
Safety Case for ML Component
Feedback and Iterate


[H] ML Safety Requirements
[P] Verification Data
[V] ML Model
[BB] ML Verification Argument Pattern
13. Verify ML Model
14. Instansiate ML Verification Argument Pattern
[Z] ML Verification Results
[CC] ML Verification Argument
[AA] Verification Log


G5.1
{ML safety requirement} is satisfied
G5.2
Verification of the MLM is independent of the development of the MLM
S5.1
Argument over ML verification approach
J5.1
{Justification for the verification approach that is chosen}
J
[AA] Sn5.1
{Verification Log}
C5.2
{formally specified properties}
[P] C5.1
{Verification data samples}
G5.3
{ML safety requirement} is satisfied when using verification data
G5.4
Formal verification demonstrates {formally specified properties} are satisfied
J5.2
{Rationale that the formally specified properties are a sufficient representation of the intent of the ML safety requirement in the operating environment}
J
G5.5
Verification test results show that requirement is satisfied
G5.9
The verification data used is sufficient to verify the intent of {ML safety requirement} in the operating environment}
G5.6
Test platform is sufficiently representative of operational platform
G5.7
Results of formal verification prove that the formally specified properties hold in the MLM
G5.8
The formal model is sufficiently representative of the MLM and its operating environment
[Z] Sn5.2
{Verification Test Results}
[AA] Sn5.4
{Verification Log}
[Z] Sn5.3
{Formal Verification Results}

6. Model Deployment





Integrate ML Model

- Monitor
  - Validity of key system and environment assumptions
  - output and internal states of ML model during operation
  - erroneous inputs to the ML model
    - noise and uncertainty in other components environmental uncertainty
    - adversarial behaviours
- Can use statistical techniques to check inputs are close to the training data distributions

Integrate ML Model

- System must be designed to maintain safety even in face of predicted erroneous outputs from ML model

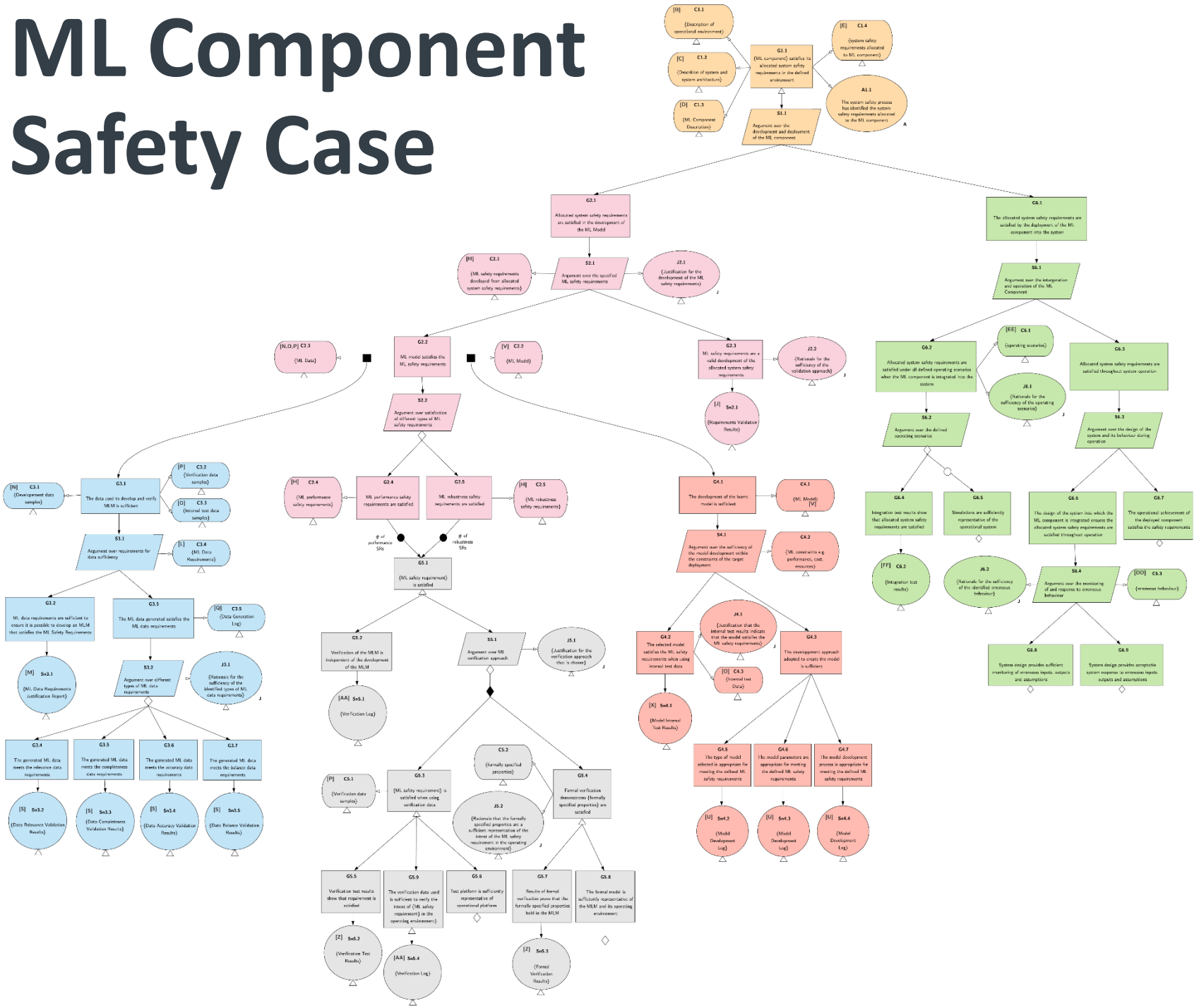
Example 46.

The ML model for pedestrian detection deployed in a self driving car has a performance requirement of 80% accuracy. Due to uncertainty in the model this performance cannot be achieved for every frame. The model uses as inputs a series of multiple images derived from consecutive image frames obtained from a camera. The presence of a pedestrian is determined by considering the result in the majority of the frames in the series. In this way the system compensates for the possible error of the model for any single image used.

Test the integration

- Following integration must check the system safety requirements are satisfied
- Requires operational scenarios against which behaviour implemented in ML can be tested
- Integration testing may include:
  - Simulation
  - Hardware in the loop
  - Shadow deployment

ML Component Safety Case



On-Going Work

- Case studies
  - Wildfire Detection
  - Sepsis Diagnosis
- Interactive Website
- Tooling
- Similar Guidance for System level

ASSURING AUTONOMY

INTERNATIONAL PROGRAMME

Funded by

Lloyd's Register Foundation