Software
Improvement
Group

(Gen)AI security & privacy controls

Developments at ISO/IEC, CEN/CENELEC and OWASP

Rob van der Veer

FEBRUARY 19 2024

Introduction

Rob van der Veer

Senior director AI, security & privacy

Software Improvement Group

- 32 years experience AI, security & privacy
- ISO/IEC:
  - 5338 - AI lifecycle (Lead author)
  - 27090 - AI security
  - 27091 - AI privacy
- Advisor ENISA, Dutch NCSC, CIP
- OWASP: SAMM, AI Exchange, ML top 10, OpenCRE.org
- CEN/CENELEC JTC21/WG 5 (EU AI Act cybersec requirements)

Software Improvement Group

rob.vanderveer@softwareimprovementgroup.com

@robvanderveer

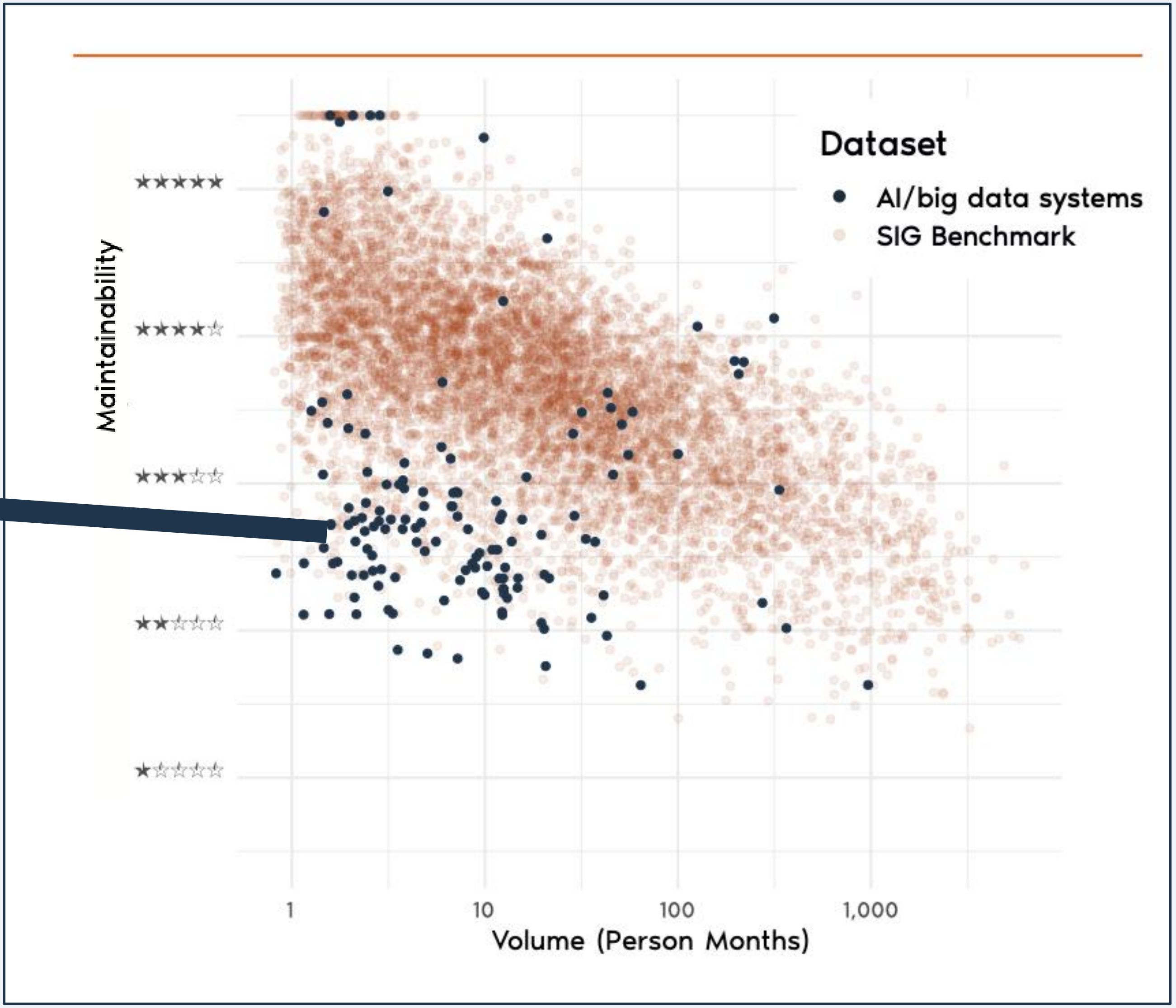
+31 6 20437187

Why ISO/IEC 5338: Software engineering best practices are often lacking in AI

AI systems built by data scientists in the lab

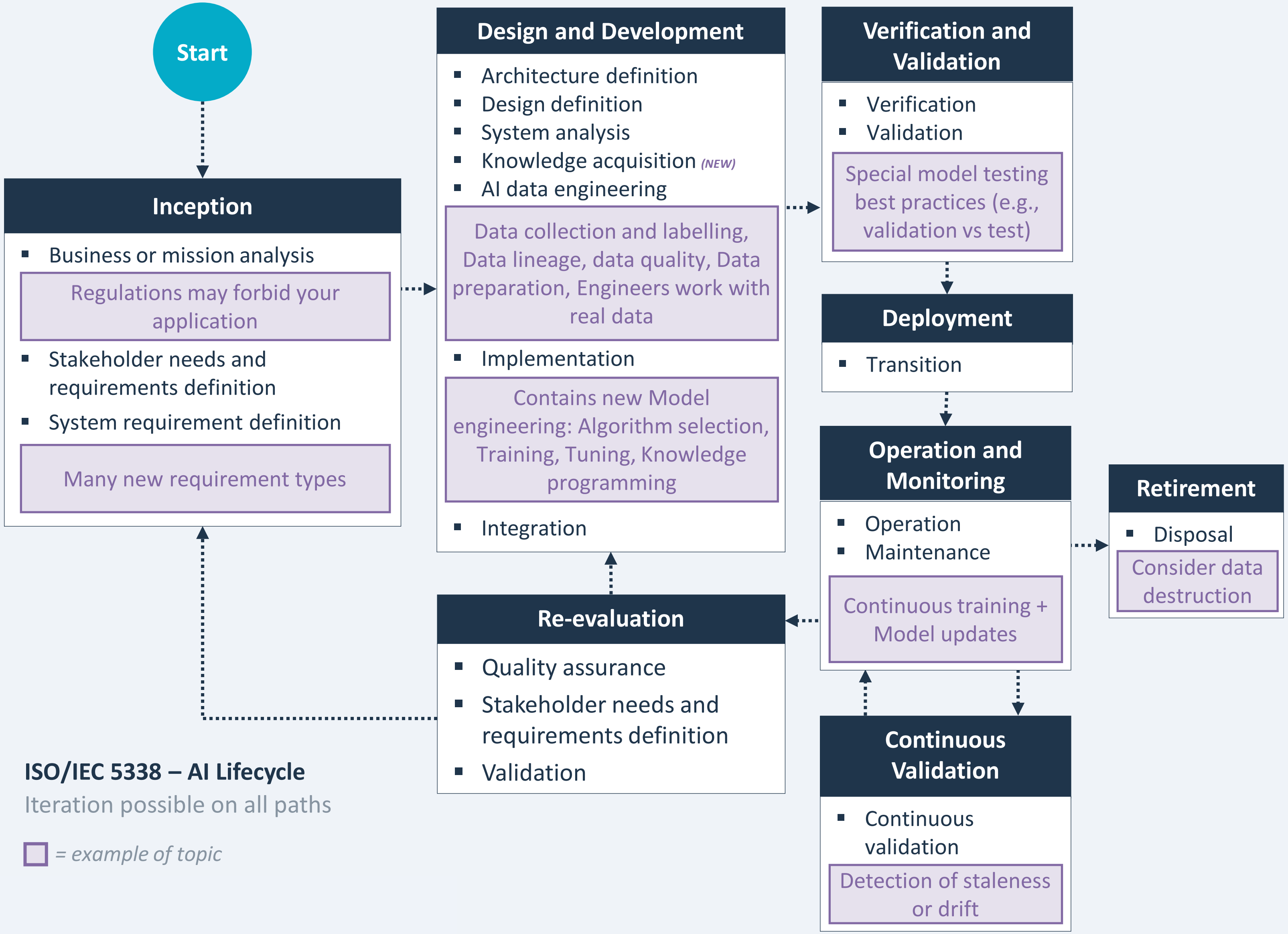
- Bad maintainability
- Zero test code
- Lack of security
- Privacy issues
- Scalability issues
- Undocumented

An accident or failure waiting to happen



Source: SIG benchmark report 2023 at https://www.softwareimprovementgroup.com/publications/2023-benchmark-report/

ISO/IEC 5338 on AI system lifecycle: AI particularities for 12207 (software lifecycle)



Start

Inception

- Business or mission analysis

Regulations may forbid your application

- Stakeholder needs and requirements definition
- System requirement definition

Many new requirement types

Design and Development

- Architecture definition
- Design definition
- System analysis
- Knowledge acquisition *(NEW)*
- AI data engineering

Data collection and labelling, Data lineage, data quality, Data preparation, Engineers work with real data

- Implementation

Contains new Model engineering: Algorithm selection, Training, Tuning, Knowledge programming

- Integration

Verification and Validation

- Verification
- Validation

Special model testing best practices (e.g., validation vs test)

Deployment

- Transition

Operation and Monitoring

- Operation
- Maintenance

Continuous training + Model updates

Retirement

- Disposal

Consider data destruction

Continuous Validation

- Continuous validation

Detection of staleness or drift

Re-evaluation

- Quality assurance
- Stakeholder needs and requirements definition
- Validation

ISO/IEC 5338 – AI Lifecycle

Iteration possible on all paths

□ *= example of topic*

Agreement processes

- Acquisition
- Supply

Monitoring and intervention is part of supply

Organizational project-enabling

- Life cycle model management
- Infrastructure management
- Portfolio management

Keep model inventory

- Human resource management

Data scientist role

- Quality management
- Knowledge management

Technical management

- Project planning
- Project assessment and control
- Decission management
- Risk management

New risks

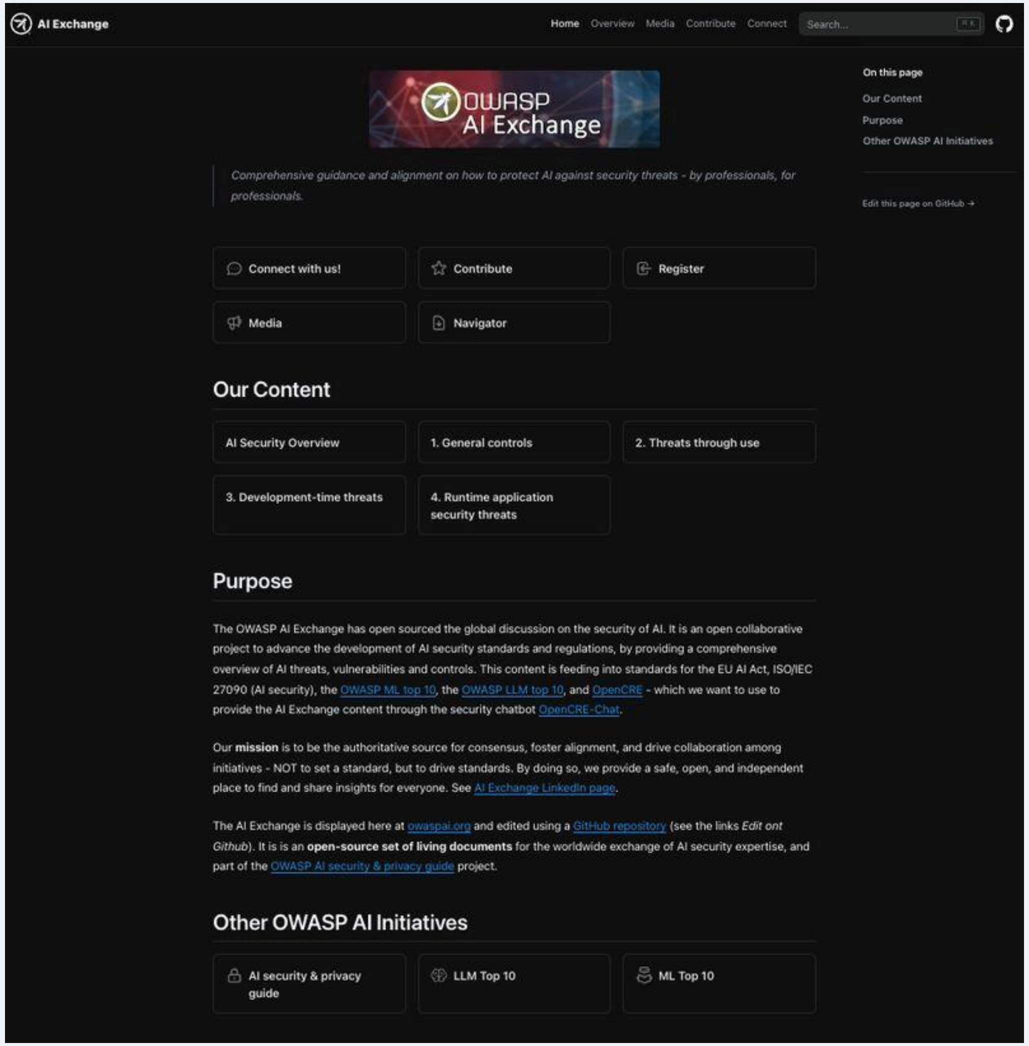
- Configuration management
- Information management

Include train data and models

- Measurement
- Quality assurance

OWASP AI Exchange at owaspai.org

- Well-received OWASP project since Q4 2022
- Open sources the AI security discussion
- Cross-geo, cross-domain, cross-industry
- Copyright and attribution-free
- Active liaisons:
  CEN/CENELEC, CSA, ISO 27090, NIST, MITRE, NCSCs, AISIC
- Content adopted verbatim by CEN/CENELEC JTC 21 for the AI Act



> *Comprehensive guidance and alignment on how to protect AI against security threats - by professionals, for professionals.*

Our Content

AI Security Overview

1. General controls

2. Threats through use

3. Development-time threats

4. Runtime application security threats

Purpose

The OWASP AI Exchange has open sourced the global discussion on the security of AI. It is an open collaborative project to advance the development of AI security standards and regulations, by providing a comprehensive overview of AI threats, vulnerabilities and controls. This content is feeding into standards for the EU AI Act, ISO/IEC 27090 (AI security), the OWASP ML top 10, the OWASP LLM top 10, and OpenCRE - which we want to use to provide the AI Exchange content through the security chatbot OpenCRE-Chat.

Our **mission** is to be the authoritative source for consensus, foster alignment, and drive collaboration among initiatives - NOT to set a standard, but to drive standards. By doing so, we provide a safe, open, and independent place to find and share insights for everyone. See AI Exchange LinkedIn page.

The AI Exchange is displayed here at owaspai.org and edited using a GitHub repository (see the links *Edit ont Github*). It is is an **open-source set of living documents** for the worldwide exchange of AI security expertise, and part of the OWASP AI security & privacy guide project.

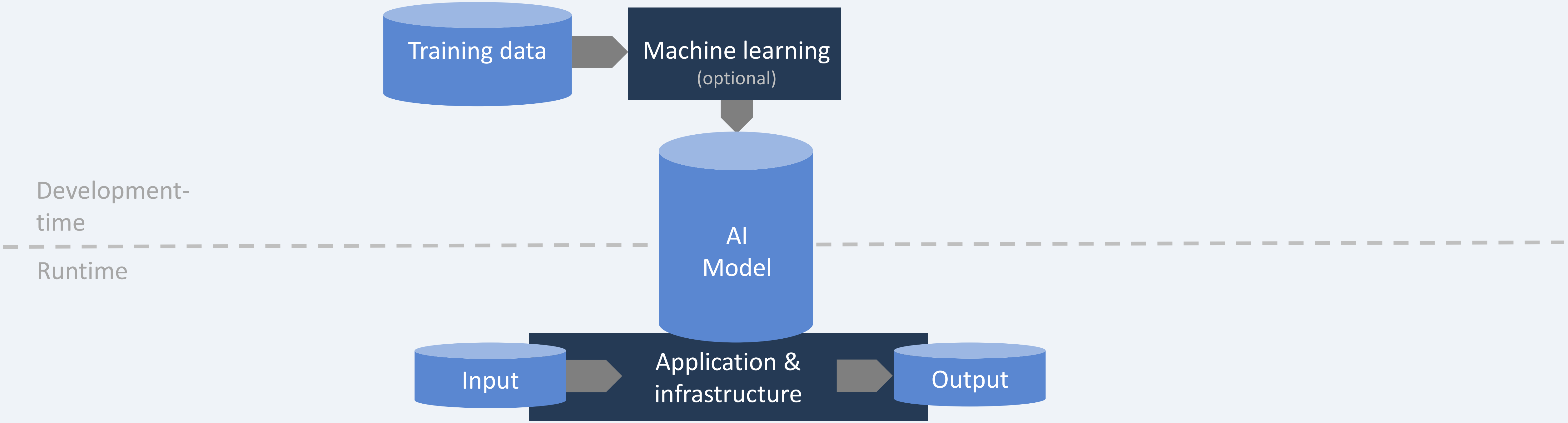
Other OWASP AI Initiatives

AI security & privacy guide

LLM Top 10

ML Top 10

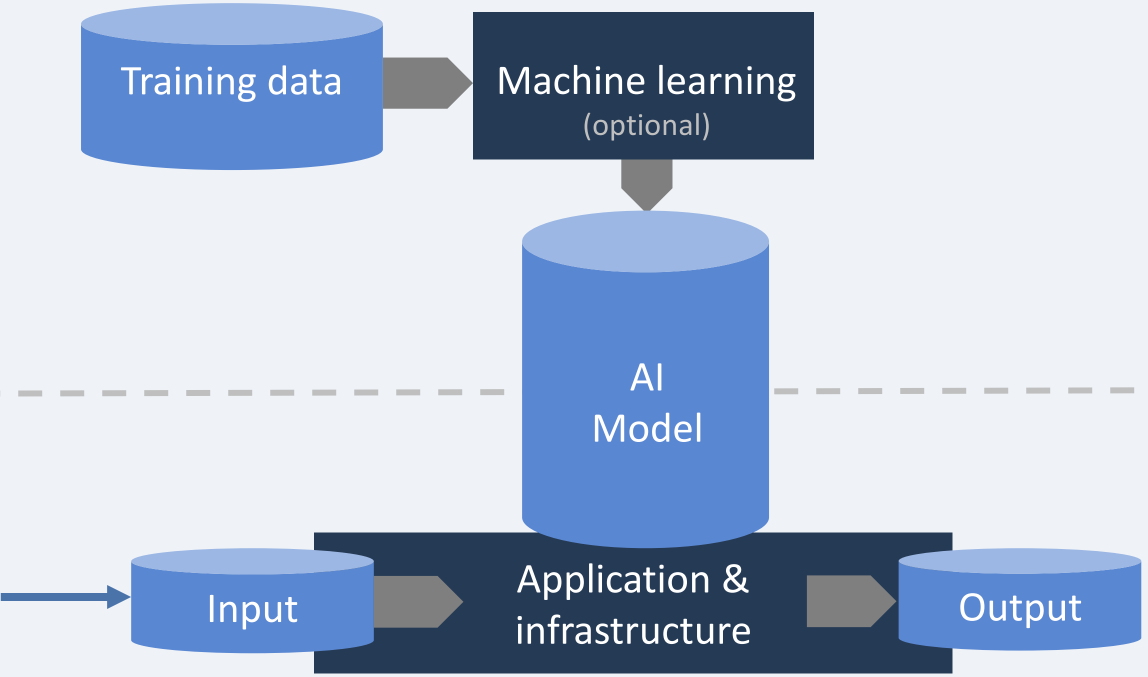
Threats framework - OWASP AI Exchange



Threats framework - OWASP AI Exchange

Development-time

Runtime



Threats through use:

- Evasion$^{(B)}$
- Model theft$^{(P)}$
- Model inversion $^{(T)}$
- Data disclosure$^{(T)}$
- Membership inference$^{(T)}$
- Denial of model service$^{(A)}$
- Prompt injection$^{(B)}$

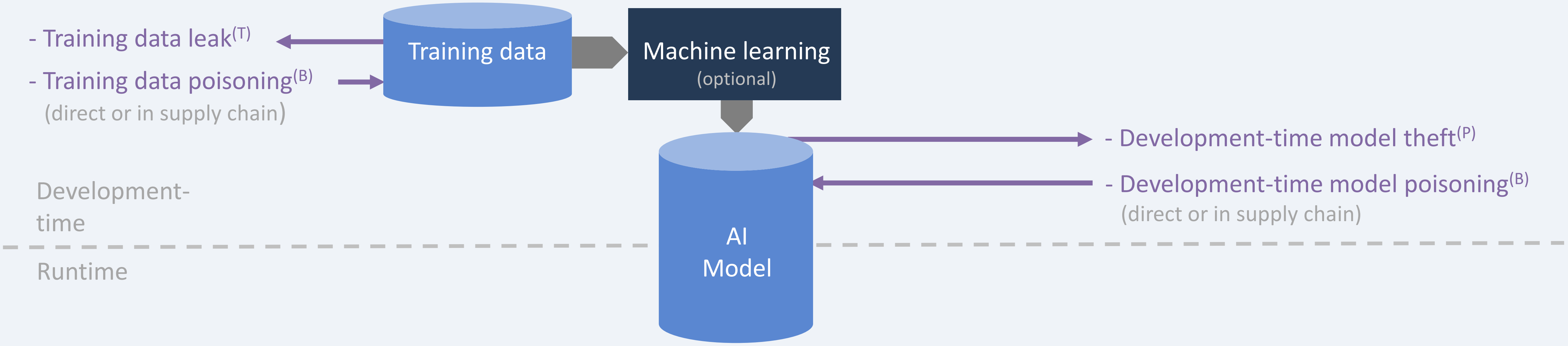
Impact legend:

(T) Train data confidentiality
(B) Model behaviour
(P) Intellectual property
(A) Availability
(L) Input confidentiality

➔ = threat

Threats framework - OWASP AI Exchange

Development-time threats



- Training data leak(T)

- Training data poisoning(B)
(direct or in supply chain)

Training data

Machine learning
(optional)

AI
Model

- Development-time model theft(P)

- Development-time model poisoning(B)
(direct or in supply chain)

Development-
time

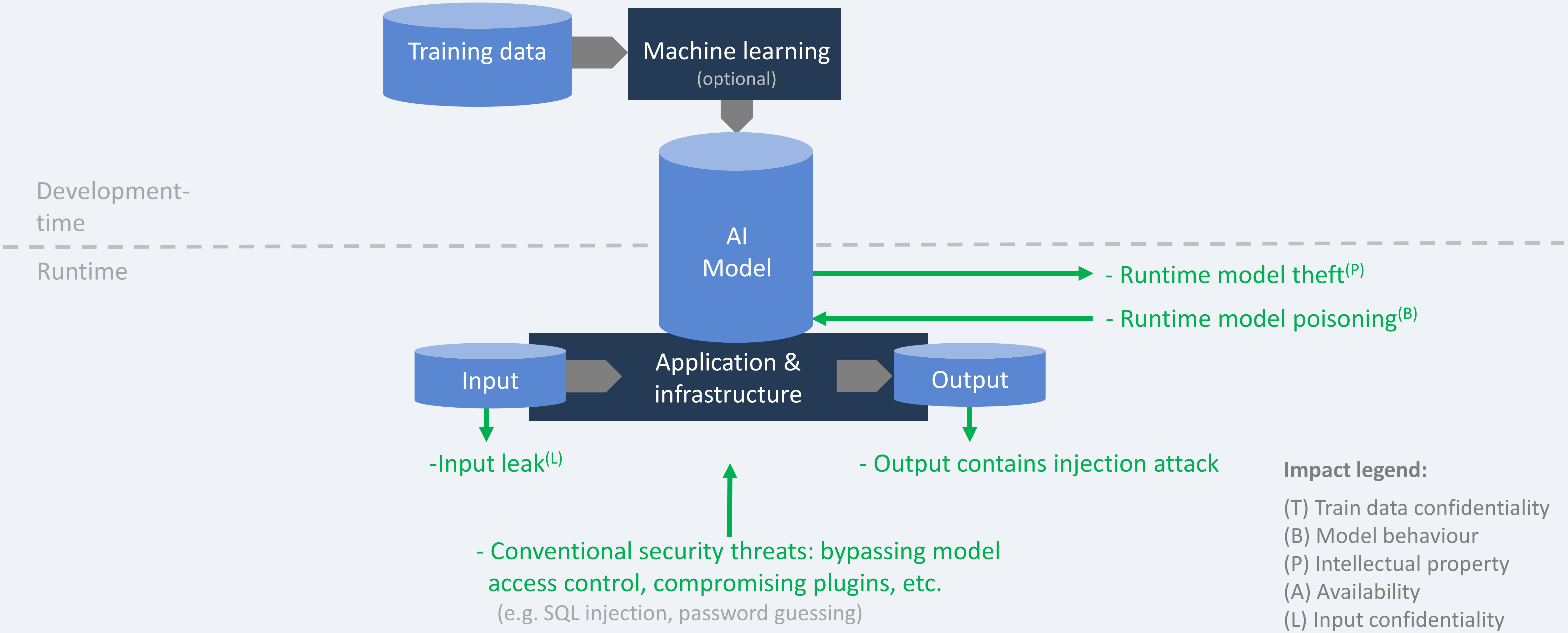
Runtime

Impact legend:

(T) Train data confidentiality
(B) Model behaviour
(P) Intellectual property
(A) Availability
(L) Input confidentiality

➔ = threat

Threats framework - OWASP AI Exchange



Training data

Machine learning
(optional)

Development-
time

Runtime

AI
Model

- Runtime model theft(P)

- Runtime model poisoning(B)

Input

Application &
infrastructure

Output

-Input leak(L)

- Output contains injection attack

- Conventional security threats: bypassing model access control, compromising plugins, etc.
(e.g. SQL injection, password guessing)

Runtime security threats

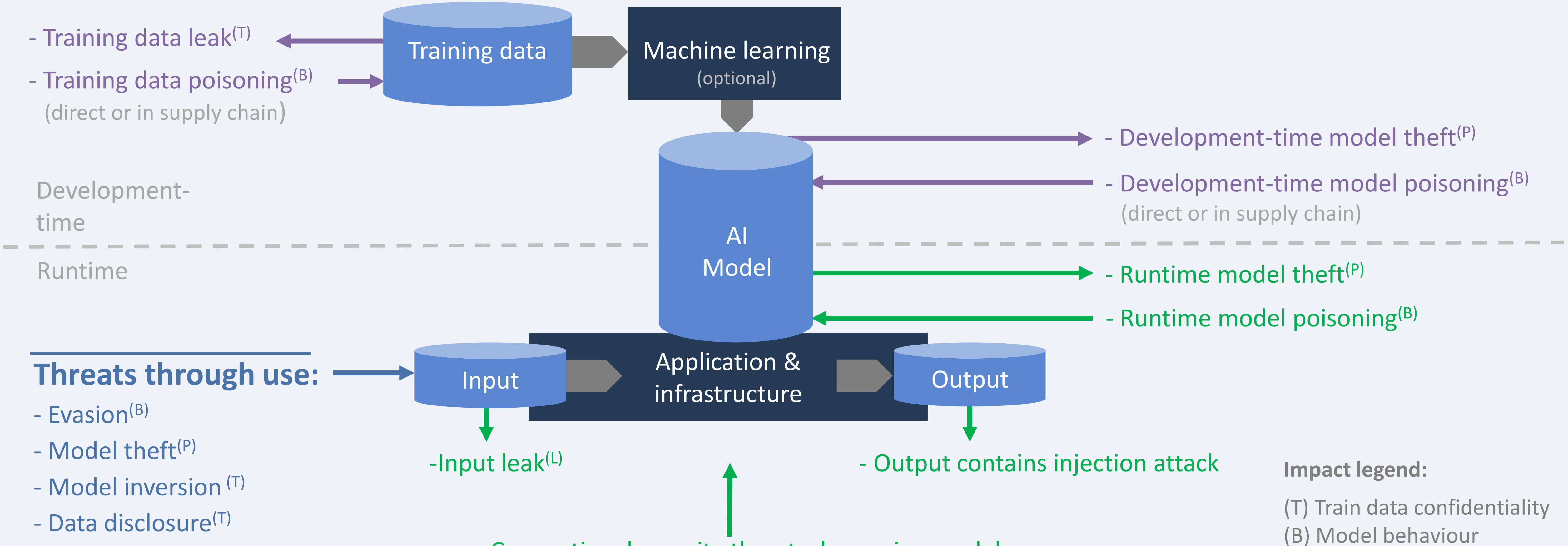
Impact legend:

(T) Train data confidentiality
(B) Model behaviour
(P) Intellectual property
(A) Availability
(L) Input confidentiality

➔ = threat

Threats framework - OWASP AI Exchange

Development-time threats



- Training data leak(T)
- Training data poisoning(B)
  (direct or in supply chain)

Training data

Machine learning
(optional)

AI Model

- Development-time model theft(P)
- Development-time model poisoning(B)
  (direct or in supply chain)

Development-time

Runtime

- Runtime model theft(P)
- Runtime model poisoning(B)

Input

Application & infrastructure

Output

-Input leak(L)

- Output contains injection attack

Threats through use:

- Evasion(B)
- Model theft(P)
- Model inversion (T)
- Data disclosure(T)
- Membership inference(T)
- Denial of model service(A)
- Prompt injection(B)

- Conventional security threats: bypassing model access control, compromising plugins, etc.
  (e.g. SQL injection, password guessing)

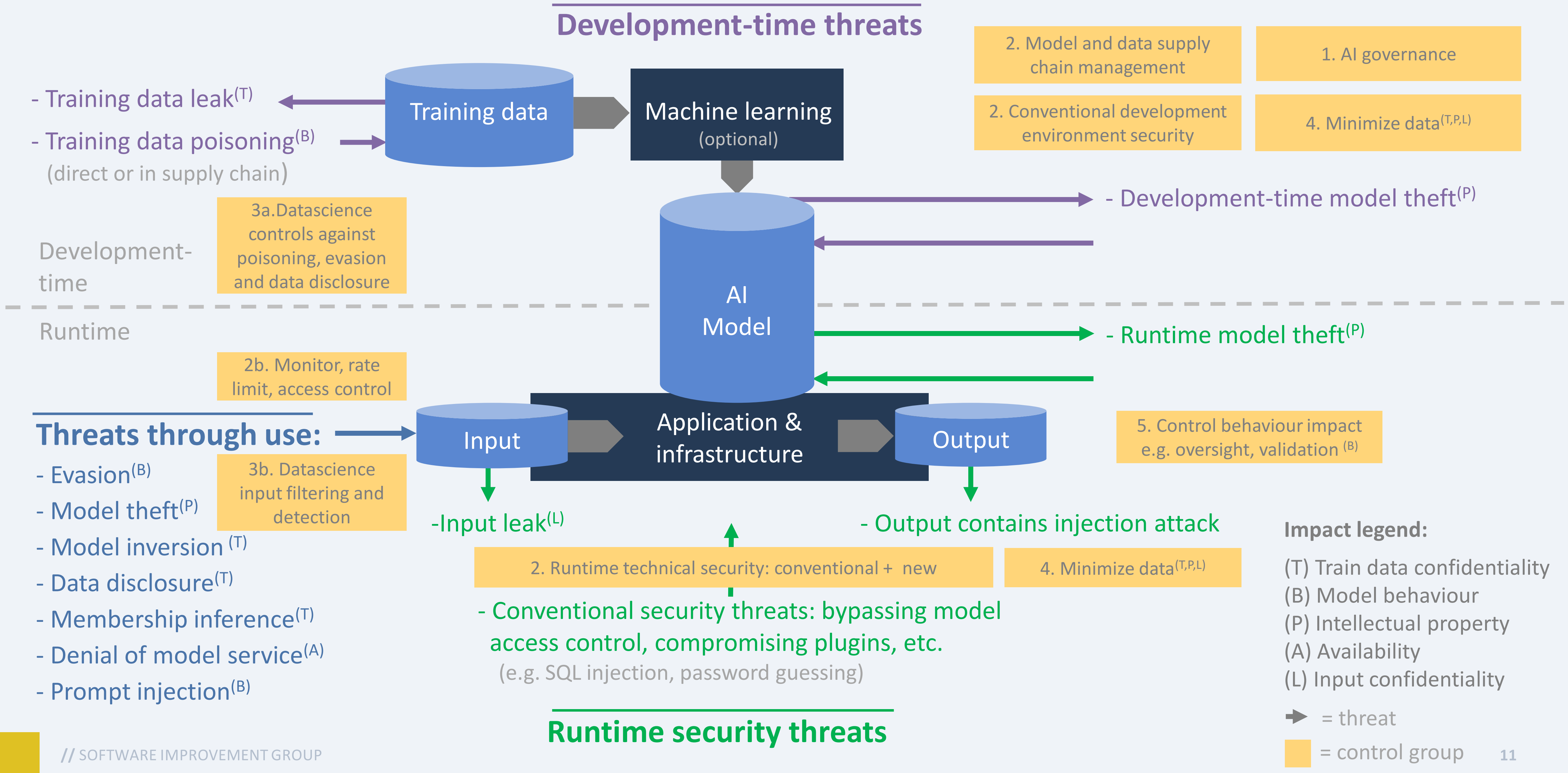
Runtime security threats

Impact legend:

(T) Train data confidentiality
(B) Model behaviour
(P) Intellectual property
(A) Availability
(L) Input confidentiality

➔ = threat

Threats framework + control groups - OWASP AI Exchange



Development-time threats

- Training data leak(T)

- Training data poisoning(B)
(direct or in supply chain)

Training data

Machine learning
(optional)

2. Model and data supply chain management

1. AI governance

2. Conventional development environment security

4. Minimize data(T,P,L)

- Development-time model theft(P)

3a.Datascience controls against poisoning, evasion and data disclosure

Development-time

AI Model

Runtime

- Runtime model theft(P)

2b. Monitor, rate limit, access control

Threats through use:

- Evasion(B)
- Model theft(P)
- Model inversion (T)
- Data disclosure(T)
- Membership inference(T)
- Denial of model service(A)
- Prompt injection(B)

3b. Datascience input filtering and detection

Input

Application & infrastructure

Output

5. Control behaviour impact e.g. oversight, validation (B)

-Input leak(L)

- Output contains injection attack

2. Runtime technical security: conventional + new

4. Minimize data(T,P,L)

- Conventional security threats: bypassing model access control, compromising plugins, etc.
(e.g. SQL injection, password guessing)

Runtime security threats

Impact legend:

(T) Train data confidentiality
(B) Model behaviour
(P) Intellectual property
(A) Availability
(L) Input confidentiality

➡ = threat

▨ = control group

Control groups with individual controls - OWASP AI Exchange

1. **Governance processes**: AI risk management, Information security management, AI (secure) software engineering:
   (AIPROGRAM, SECPROGRAM, DEVPROGRAM, SECDEVPROGRAM, CHECKCOMPLIANCE, SECEDUCATE)
2. **Technical IT security controls**:
   a. Apply conventional IT security controls for **AI-specific parts**:
      - Development-time: model & data storage, model & data supply chain, data science documentation
        (DISCRETE, DEVDATAPROTECT, DEVSECURITY, SEGREGATEDATA, SUPPLYCHAINMANAGE)
      - Runtime: model storage, model use and model IO
        (RUNTIMEMODELINTEGRITY, RUNTIMEMODELIOINTEGRITY, RUNTIMEMODELCONFIDENTIALITY, MODELINPUTCONFIDENTIALITY, ENCODEMODELOUTPUT, LIMITRESOURCES)

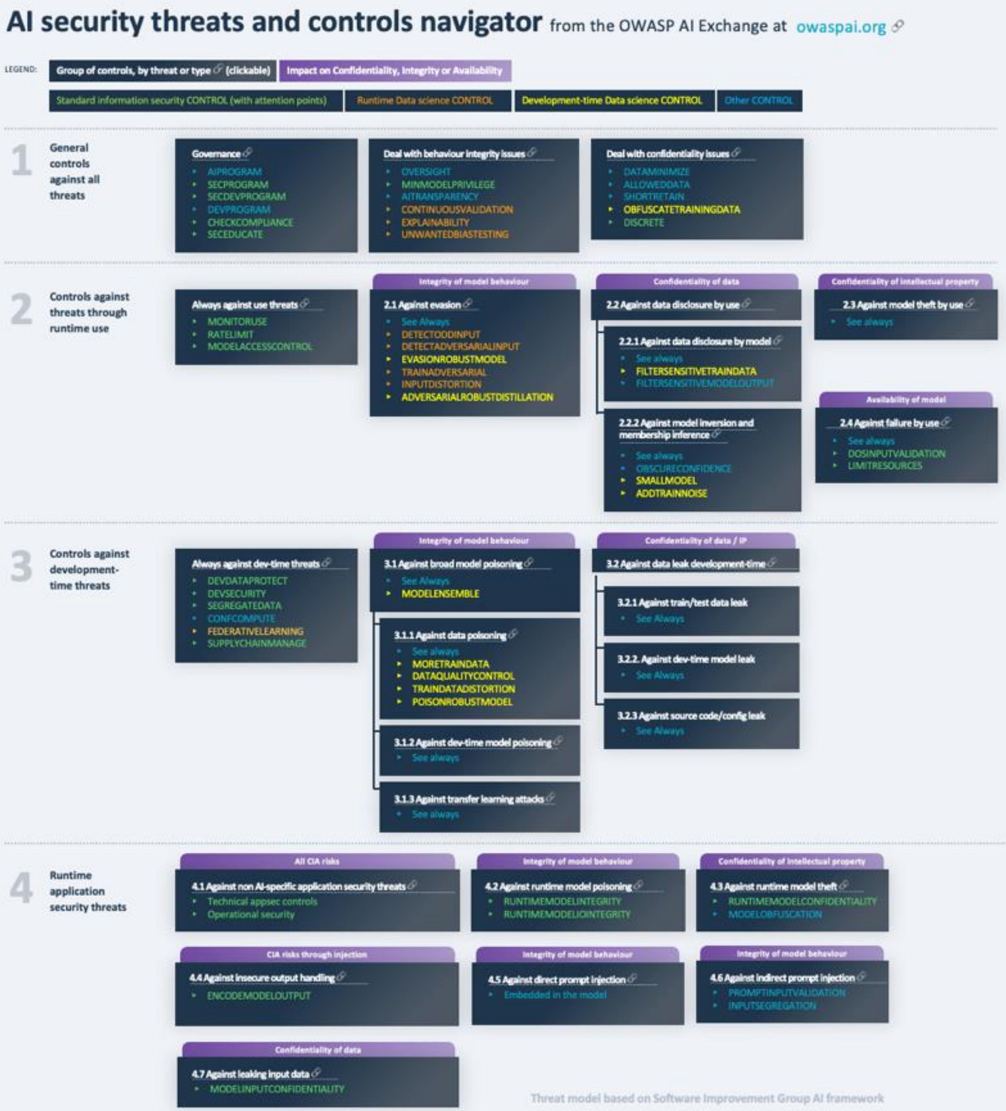
   b. **Adapt conventional** IT security controls (rate limit, monitor)
   (MONITORUSE, MODELACCESSCONTROL, RATELIMIT)

   c. **Adopt new** IT security controls
   (CONFCOMPUTE, MODELOBFUSCATION, PROMPTINPUTVALIDATION, INPUTSEGREGATION)
3. **Datascience security controls:**

   a. **Development-time** controls when developing the model
   (DATAQUALITYCONTROL, FEDERATIVELEARNING, EVASIONROBUSTMODEL, POISIONROBUSTMODEL, TRAINADVERSARIAL, TRAINDATADISTORTION, ADVERSARIALROBUSTDISTILLATION, FILTERSENSITIVETRAINDATA, MODELENSEMBLE, MORETRAINDATA, SMALLMODEL)

   b. **Runtime** controls when running the model:
   (DETECTODDINPUT, DETECTADVERSARIALINPUT, DOSINPUTVALIDATION, INPUTDISTORTION, FILTERSENSITIVEMODELOUTPUT, OBSCURECONFIDENCE)
4. **Limit the amount of data** and the time it is stored
   (DATAMINIMIZE, ALLOWEDDATA, SHORTRETAIN, OBFUSCATETRAININGDATA)
5. **Limit the effect** of unwanted model behaviour (e.g. oversight, validation)
   (OVERSIGHT, LEASTMODELPRIVILEGE, CONTINUOUSVALIDATION, AITRANSPARENCY, EXPLAINABILITY, UNWANTEDBIASTESTING)

Follow the AI Exchange & LLM top 10 - owaspai.org & LLMtop10.com



AI security threats and controls navigator from the OWASP AI Exchange at owaspai.org

LEGEND:
Group of controls, by threat or type (clickable)
Impact on Confidentiality, Integrity or Availability
Standard information security CONTROL (with attention points)
Runtime Data science CONTROL
Development-time Data science CONTROL
Other CONTROL

1 General controls against all threats

Governance
- AIPROGRAM
- SECPROGRAM
- SECDEVPROGRAM
- DEVPROGRAM
- CHECKCOMPLIANCE
- SECEDUCATE

Deal with behaviour integrity issues
- OVERSIGHT
- MINMODELPRIVILEGE
- AITRANSPARENCY
- CONTINUOUSVALIDATION
- EXPLAINABILITY
- UNWANTEDBIASTESTING

Deal with confidentiality issues
- DATAMINIMIZE
- ALLOWEDDATA
- SHORTRETAIN
- OBFUSCATETRAININGDATA
- DISCRETE

2 Controls against threats through runtime use

Always against use threats
- MONITORUSE
- RATELIMIT
- MODELACCESSCONTROL

Integrity of model behaviour

2.1 Against evasion
- See Always
- DETECTODDINPUT
- DETECTADVERSARIALINPUT
- EVASIONROBUSTMODEL
- TRAINADVERSARIAL
- INPUTDISTORTION
- ADVERSARIALROBUSTDISTILLATION

Confidentiality of data

2.2 Against data disclosure by use

2.2.1 Against data disclosure by model
- See always
- FILTERSENSITIVETRAINDATA
- FILTERSENSITIVEMODELOUTPUT

2.2.2 Against model inversion and membership inference
- See always
- OBSCURECONFIDENCE
- SMALLMODEL
- ADDTRAINNOISE

Confidentiality of intellectual property

2.3 Against model theft by use
- See always

Availability of model

2.4 Against failure by use
- See always
- DOSINPUTVALIDATION
- LIMITRESOURCES

3 Controls against development-time threats

Always against dev-time threats
- DEVDATAPROTECT
- DEVSECURITY
- SEGREGATEDATA
- CONFCOMPUTE
- FEDERATIVELEARNING
- SUPPLYCHAINMANAGE

Integrity of model behaviour

3.1 Against broad model poisoning
- See Always
- MODELENSEMBLE

3.1.1 Against data poisoning
- See always
- MORETRAINDATA
- DATAQUALITYCONTROL
- TRAINDATADISTORTION
- POISONROBUSTMODEL

3.1.2 Against dev-time model poisoning
- See always

3.1.3 Against transfer learning attacks
- See always

Confidentiality of data / IP

3.2 Against data leak development-time

3.2.1 Against train/test data leak
- See Always

3.2.2. Against dev-time model leak
- See Always

3.2.3 Against source code/config leak
- See Always

4 Runtime application security threats

All CIA risks

4.1 Against non AI-specific application security threats
- Technical appsec controls
- Operational security

Integrity of model behaviour

4.2 Against runtime model poisoning
- RUNTIMEMODELINTEGRITY
- RUNTIMEMODELIOINTEGRITY

Confidentiality of intellectual property

4.3 Against runtime model theft
- RUNTIMEMODELCONFIDENTIALITY
- MODELOBFUSCATION

CIA risks through injection

4.4 Against insecure output handling
- ENCODEMODELOUTPUT

Integrity of model behaviour

4.5 Against direct prompt injection
- Embedded in the model

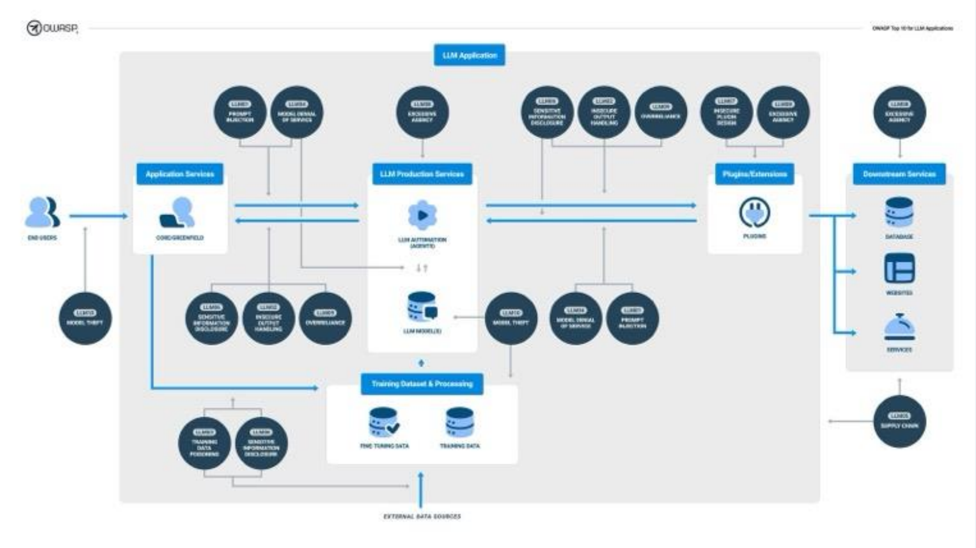
Integrity of model behaviour

4.6 Against indirect prompt injection
- PROMPTINPUTVALIDATION
- INPUTSEGREGATION

Confidentiality of data

4.7 Against leaking input data
- MODELINPUTCONFIDENTIALITY

Threat model based on Software Improvement Group AI framework



AI-specific?	Lifecycle	Attack surface	Threat	Asset	Impacted	Unwanted result
AI	Runtime	Model use (provide input/ read output)	Direct prompt injection	Model behaviour	Integrity	Manipulated unwanted model behaviour causes wrong decisions leading to business financial loss, misbehaviour going undetected, reputational damage, legal and compliance issues, operational disruption, customer dissatisfaction and churn, reduced empoloyee morale, incorrect strastegic decisions, liabiity issues, personal damage and safety issues
			Indirect prompt injection			
			Evasion (e.g. adversarial examples)			
		Break into deployed model	Runtime model poisoning (reprogramming)			
	Development	Engineering environment	Development time model poisoning			
			Data poisoning of train/finetune data			
		Supply chain	Obtain poisoned foundation model (transfer learning attack)			
			Obtain poisoned data to train/finetune			
	Runtime	Model use	Unwanted disclosure in model output	Train data	Confidentiality	Leaking sensitive data can cause costs from fines and legal fees and remediation effort, loss of business through customer churn, reputation damage, loss of competitive advantage in case of trade secrets, operational disruption, impacted business relationships, and employee morale
			Model inversion / Membership inference			
	Development	Engineering environment	Train data leaks			
	Runtime	Model use	Model theft through by use (input-output harvesting)	Model intellectual property	Confidentiality	If attackers can copy a model, the investment in the model is devalued caused by loss of competitive advantage, plus a copy can help craft (evasion) attacks
		Break into deployed model	Runtime model theft			
	Development	Engineering environment	Development time model parameter leak			
	Runtime	Model use	System failure by use (model resource depletion)	Model behaviour	Availability	The model is not available, leading to business continuity issues, or safety problems
	Runtime	All IT	Model input leak	Model input data	Confidentiality	Sensitive data in model input leaks. E.g. an LLM prompt with a sensitive question, enhanced with retrieved company secrets
	Runtime	All IT	Model output contains injection attack	Any asset	C, I, A	Injection attack (from model output) causes harm
Generic	Runtime	All IT	Generic runtime security attack	Any asset	C, I, A	Generic runtime security attack causes harm (includes social engineering/phishing)
	Development	All IT	Generic supply chain attack	Any asset	C, I, A	Generic supply chain security attack causes harm (e.g. vulnerability in a component)

How about privacy controls?

Four types of privacy controls:

1. AI security controls to **protect personal data**
2. AI security controls to **minimize data**
3. AI security controls to **limit impact** of model behaviour
4. **Additional controls to protect individual privacy rights:**
   - Validate purpose (e.g. repurposing personal data)
   - Arrange consent
   - Control unwanted bias
   - Provide transparency/explanation
   - Attain data accuracy & updating
   - Provide features to correct, access, erase, and object

Source: OWASP AI Security & privacy guide

Software
Improvement
Group.com

GETTING SOFTWARE RIGHT FOR A HEALTHIER DIGITAL WORLD