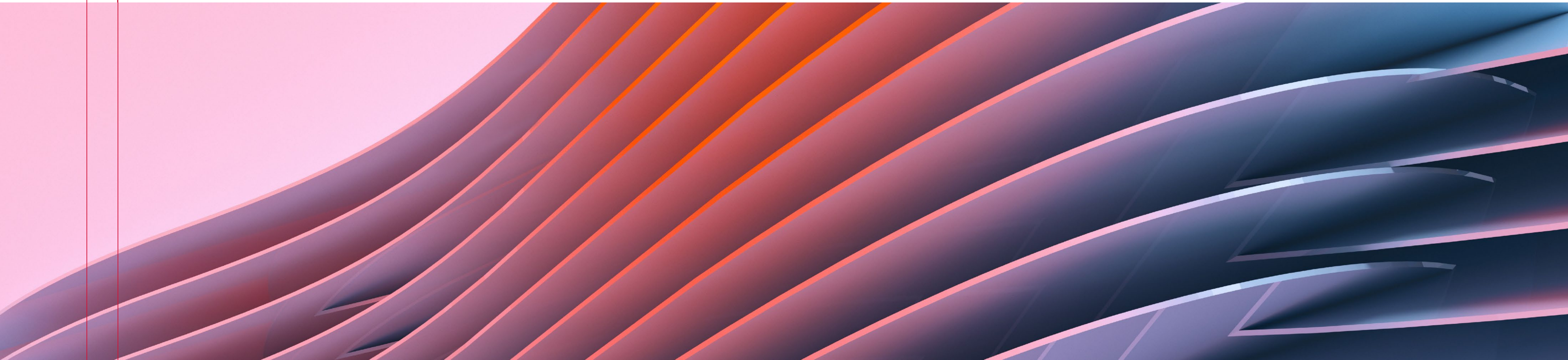
Kilpatrick

ACC Association of Corporate Counsel
NATIONAL CAPITAL REGION

SEPTEMBER 10, 2025

The Law of the Machine (Learning): Solving Complex AI Challenges



Speakers

Michael Breslin
Partner, Complex Commercial Litigation
Kilpatrick

Meghan Farmer
Partner, Technology, Privacy, & Cybersecurity
Kilpatrick

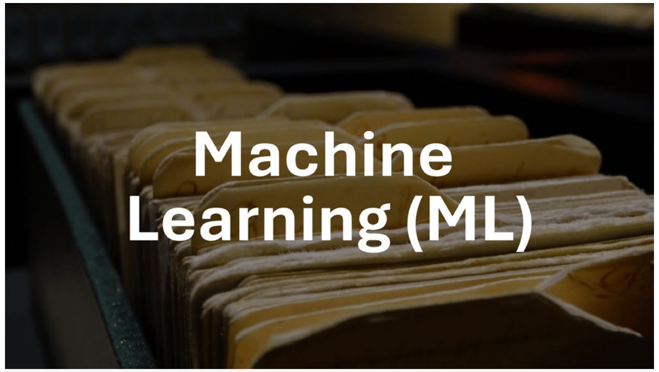
Rome Perlman
Associate General Counsel
National Student Clearinghouse

Greg Silberman
Partner, Technology, Privacy, & Cybersecurity
Kilpatrick

```
        csv_path = bpath.abspath(draw_dat.csv_export_path)
    pset = psys.settings
    particles = [p for p in psys.particles] if pset.type == 'ALIVE']
    filenames = []
    if pset.render_type == "OBJECT":
        dupli_ob = pset.dupli_object
        if dupli_ob is not None and draw_dat.instances_write_dupli:
            filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
            if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
                filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
            filepath = "".join(filepath)
            dupli_world = dupli_ob.matrix_world.copy()
            transl_inv = Matrix.Translation(-dupli_world.translation)
            dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
            filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
            dupli_ob.matrix_world = dupli_world
            obj.matrix_world = Matrix.Identity(4)
            writeObject(context, instance_filepath, [obj])
            obj.matrix_world = obj_world
    else:
        WARNING("Invalid datatype '%s'" % pset.render_type)
        return
    try:
        csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
        fh = open(csv_file, "w")
        for p in particles:
            rot = Quaternion.to_matrix(p.rotation).to_4x4()
            if (pset.type == "HAIR"):
                h1 = p.hair_keys[0].co
                h2 = p.hair_keys[-1].co
                loc = Matrix.Translation(h1)
                scale = Matrix.Scale((h2 - h1).length, 4)
                rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
            else:
                loc = Matrix.Translation(p.location)
                scale = Matrix.Scale(p.size, 4)
            t = loc * rot * scale
            t = emitter.matrix_world * t if pset.type == "HAIR" else t
            writeTransform(oct_t[0] * t * oct_t[1], fh)
        fh.close()
        filenames.append(csv_file)
        return filenames
    except IOError as err:
        raise ExportException(msg)
```

Setting the Stage

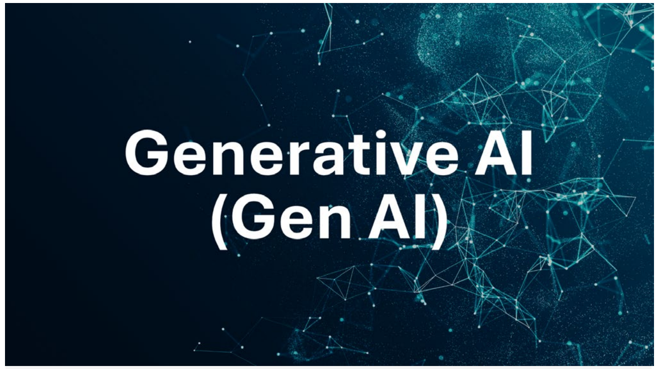
Artificial Intelligence



Machine Learning (ML)

Learns patterns from historical data to make predictions or classifications (e.g., "fraud" vs. "legitimate" transaction).

Operates only when triggered by data inputs or scheduled runs; does not initiate tasks or adapt strategies on its own.



Generative AI (Gen AI)

Generates new, synthetic content (e.g., text, images, synthetic data) that resembles or extrapolates from training data.

Responds to prompts with creative outputs, but does not independently monitor, act, or orchestrate multi-step processes.

Agentic AI

Acts autonomously toward goals, making ongoing decisions and adapting strategies in real time (persistent, goal-driven behavior).

Can autonomously sequence and execute multi-step actions, collaborate across systems, and self-improve without explicit human prompting.

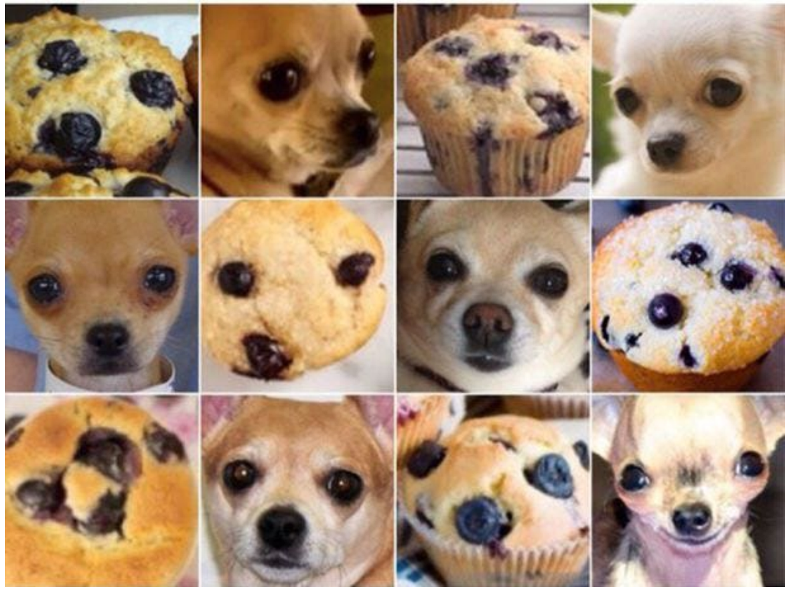
Predictive / Extractive AI vs. Generative AI

Generative artificial intelligence uses machine learning to create content, not just extract, sort, find, or make decisions about content.

Predictive, Extractive AI

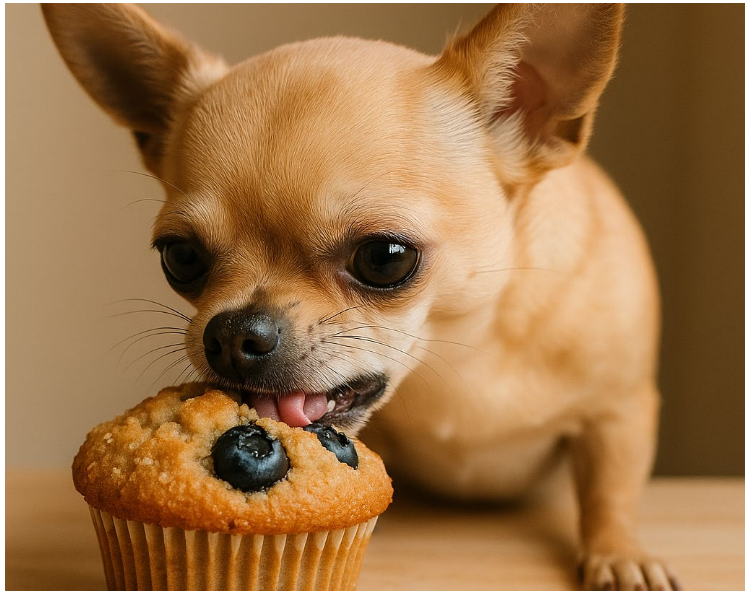
“Chihuahua or Muffin?”

Machine Learning → Pattern Recognition



Generative AI

“Create an image of a Chihuahua eating a muffin”



Data is More Than Just Fuel

- Foundation of Model Performance
- Driver of Continuous Improvement
- Source of Context and Meaning
- Anchor for Governance and Compliance
- Key to Trust and Risk Mitigation

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" else t
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

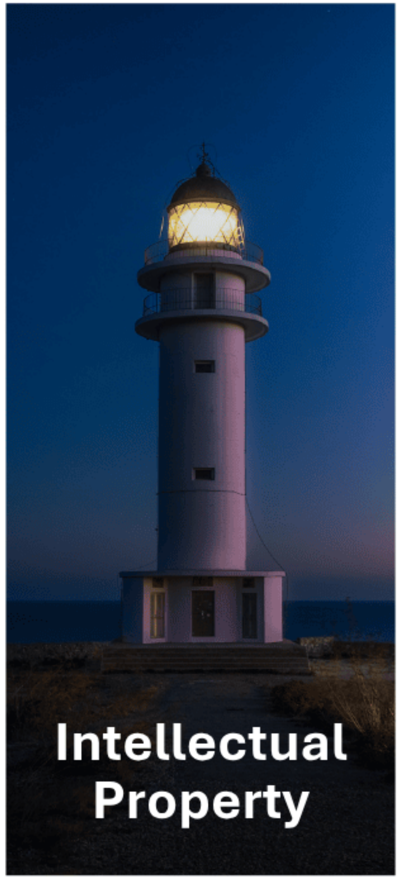
Regulatory Landscape

AI Regulatory Landscape

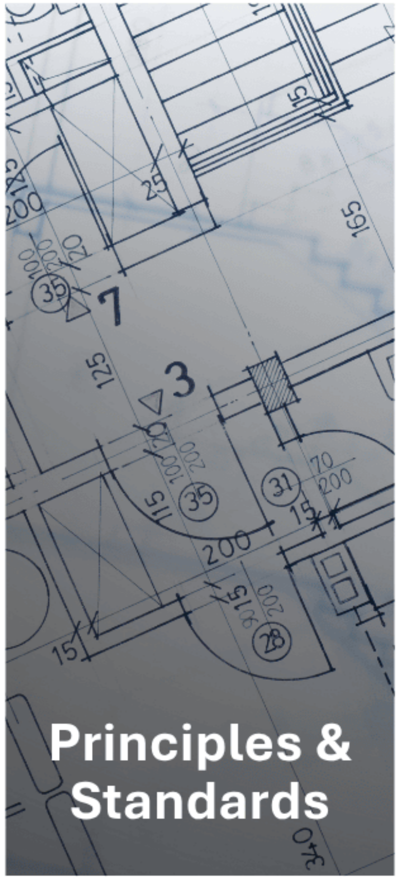
Consumer Protections

AI Specific Regulation

Privacy & Security



Intellectual Property



Principles & Standards

Federal Regulation of AI

- Sector-Based Oversight
- Anti-Discrimination via Existing Laws
- General Agency Guidance, Not Role-Specific Duties
- Limited Consumer Rights in AI Context
- Policy-Led Tech Regulation

AI Specific Regulation

AI Specific

- Colorado AI Act (eff. 2/1/26)
- CA AI Transparency Act (eff. 1/1/26)
- Utah Transparency Law (eff. 5/1/24)

AI in Employment

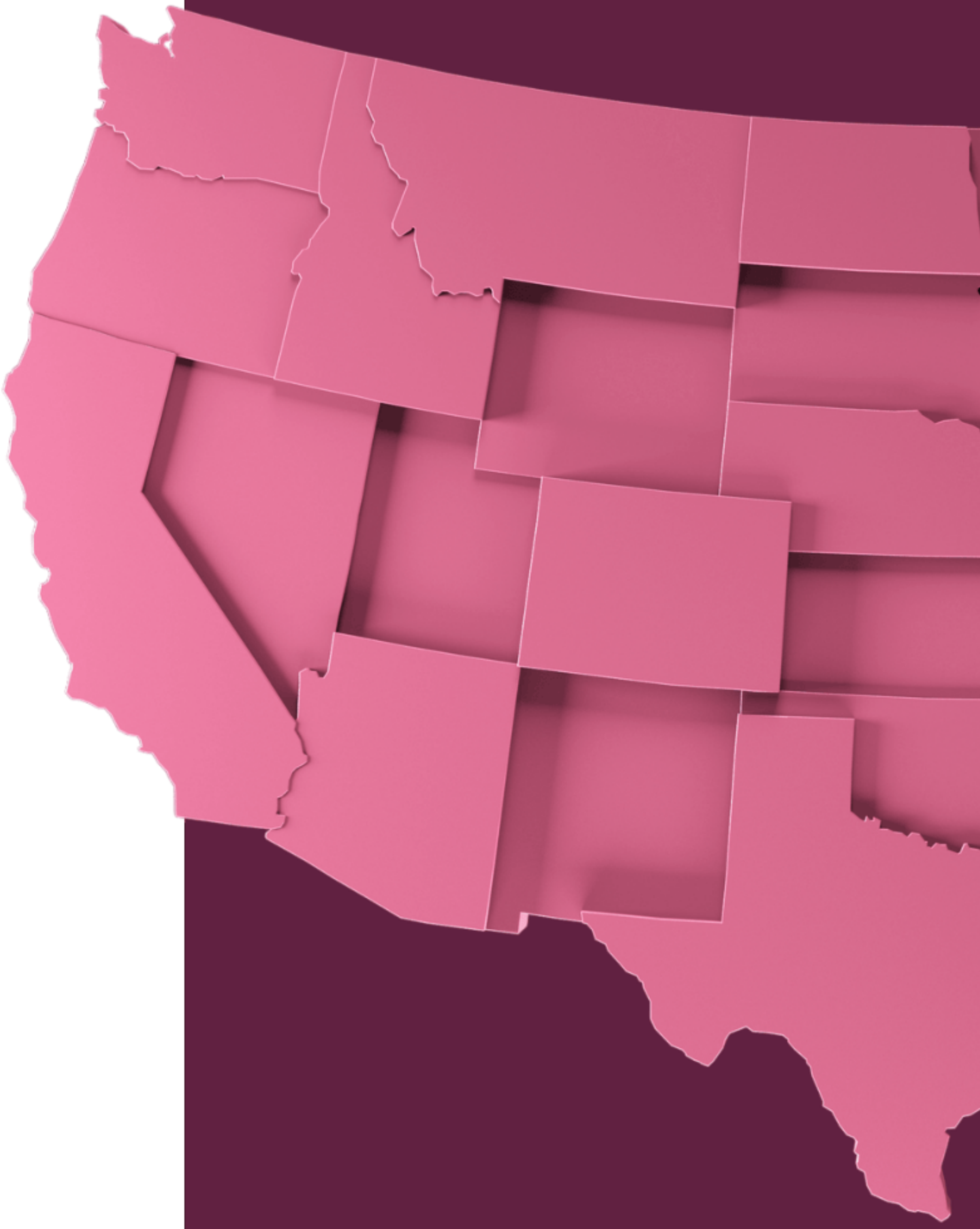
- NY Local Law 144
- Illinois AI Video Interview Act

AI in the Future

- Many states have bills that are making their way through the legislative process

State Regulation of AI

- Focus on High-Impact Use
- Preventing Discrimination
- Role-Based Obligations
- Consumer Rights Protections
- Targeted Tech Regulation



The EU AI Act in a Nutshell

- **Risk-based law** covering providers, deployers, importers, and distributors of AI used in the EU, with extraterritorial reach.
- **Four tiers:** **prohibited** practices; **high-risk** systems with mandatory controls; **limited-risk** systems with transparency duties; **minimal-risk** systems with voluntary measures.
- **High-risk obligations:** risk-management and data-quality controls; technical documentation and logging; human oversight; accuracy, robustness, and cybersecurity; post-market monitoring, incident reporting, CE marking, and EU database registration.
- **GPAI / foundation models:** baseline documentation and copyright-compliance duties; **systemic-risk** models face additional evaluations, adversarial testing, cybersecurity, and incident reporting to the EU AI Office.
- **Transparency duties:** disclose AI interaction, label synthetic or deepfake content, and inform users about biometric categorization or emotion recognition (subject to strict limits).
- **Governance and enforcement:** EU AI Office and national authorities oversee compliance; conformity assessments; significant fines tied to global turnover and possible product withdrawal.

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
  dupli_ob = pset.dupli_object
  if dupli_ob is not None and draw_dat.instances_write_dupli:
      filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
      if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
          filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
      filepath = "".join(filepath)
      dupli_world = dupli_ob.matrix_world.copy()
      transl_inv = Matrix.Translation(-dupli_world.translation)
      dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
      filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
      dupli_ob.matrix_world = dupli_world
      obj.matrix_world = Matrix.Identity(4)
      writeObject(context, instance_filepath, [obj])
      obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
  csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
  fh = open(csv_file, "w")
  for p in particles:
    rot = Quaternion.to_matrix(p.rotation).to_4x4()
    if (pset.type == "HAIR"):
        h1 = p.hair_keys[0].co
        h2 = p.hair_keys[-1].co
        loc = Matrix.Translation(h1)
        scale = Matrix.Scale((h2 - h1).length, 4)
        rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
    else:
        loc = Matrix.Translation(p.location)
        scale = Matrix.Scale(p.size, 4)
    t = loc * rot * scale
    t = emitter.matrix_world * t if pset.type == "HAIR" else t
    writeTransform(oct_t[0] * t * oct_t[1], fh)
  fh.close()
  filenames.append(csv_file)
  return filenames
except IOError as err:
  raise ExportException(msg)
```

AI Governance and Responsible AI

Frameworks and Technical Standards

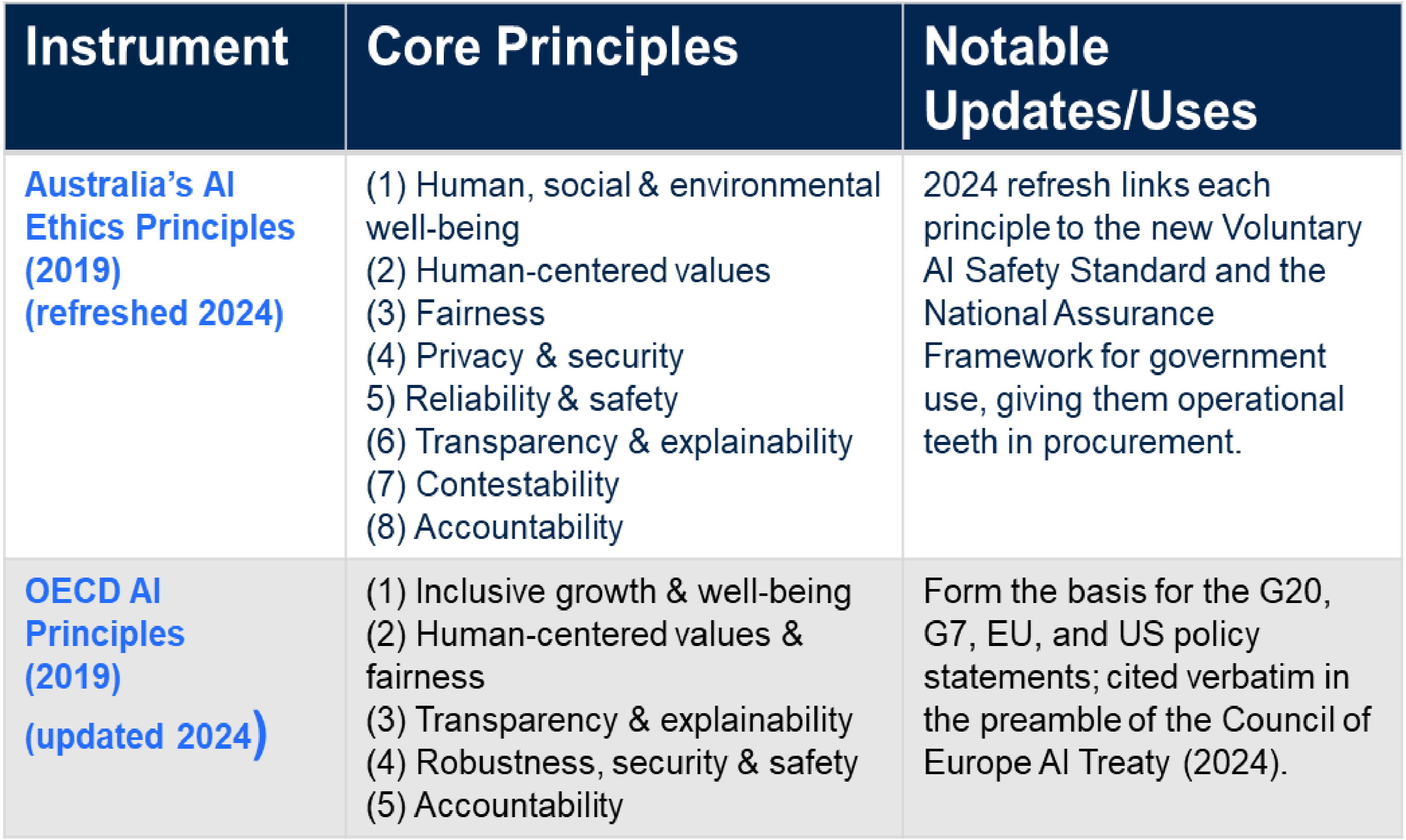
Framework	Why it Matters	Lifecycle Stages
NIST AI Risk Management Framework 1.0 (Jan 2023)	Provides a common language **(“Map-Govern-Measure-Manage”)** used by the federal agencies and regulators when they talk about “trustworthy AI”.	Design, development, deployment, monitoring
ISO/IEC 42001:2023 (AI Management System)	First certifiable AI-governance standard; aligns neatly with **ISO 27001 controls**	End-to-end governance
ISO/IEC 23894:2023 (Risk Management Guidance)	Offers detailed risk taxonomy and control suggestions and that can be integrated with existing control frameworks.	Risk assessment, validation, change management
IEEE 7000-series (Ethics-by-Design)	Voluntary. Useful as a benchmark for documenting value-based design choices (fairness, autonomy, agency).	Early design

AI Ethics and Governance Principles

2021 UNESCO
Recommendation on the Ethics of Artificial Intelligence

2022 US White House
Blueprint for an AI Bill of Rights

2024 Frontier Model Forum (FMF)
Frontier AI Safety Commitments



Instrument	Core Principles	Notable Updates/Uses
Australia's AI Ethics Principles (2019) (refreshed 2024)	(1) Human, social & environmental well-being (2) Human-centered values (3) Fairness (4) Privacy & security 5) Reliability & safety (6) Transparency & explainability (7) Contestability (8) Accountability	2024 refresh links each principle to the new Voluntary AI Safety Standard and the National Assurance Framework for government use, giving them operational teeth in procurement.
OECD AI Principles (2019) (updated 2024)	(1) Inclusive growth & well-being (2) Human-centered values & fairness (3) Transparency & explainability (4) Robustness, security & safety (5) Accountability	Form the basis for the G20, G7, EU, and US policy statements; cited verbatim in the preamble of the Council of Europe AI Treaty (2024).

Responsible AI: Accountability and Transparency

- **Accountability Goals: All AI systems must…**
  - Be assessed using impact assessments;
  - Be reviewed to identify systems that may have a significant adverse impact on people, organizations, and society;
  - Be subject to appropriate data governance and management practices;
  - Include capabilities that support informed human oversight and control.
- **Transparency Goals: The Company must…**
  - Support stakeholder needs for intelligibility of system behavior;
  - Provide information about the capabilities and limitations of the system;
  - Disclose when people are interacting with an AI system or system that generates / manipulates image, audio, or video content.

Responsible AI: Fairness, Reliability, and Safety

- **Fairness Goals: AI systems...**
  - Should be designed to provide a similar quality of service across all demographic groups;
  - Should allocate resources or opportunities in a manner that minimizes disparities in outcomes across demographic groups;
  - Should describe, depict, or represent people, cultures, and society in a way that minimizes stereotyping, demeaning, or marginalization.
- **Reliability and Safety Goals: The Company must...**
  - Evaluate the operational factors which systems are expected to perform reliably and safely;
  - Design AI systems to minimize time to remediation of predictable or known failures and subject systems to ongoing monitoring and feedback.

Policy Framework

Purpose and Scope – Clearly define the objective of your AI policy. Is it focused on ethics, legal compliance, or both? Specify which AI systems the policy applies to and whether it covers all AI or specific types like machine learning.

Transparency and Explainability – Promote transparency in AI use and decision-making processes whenever possible.

Continuous Review and Updates – Regularly review and update your AI policy to reflect advancements in technology and address any emerging challenges.

Alignment with Existing Policies – Ensure your AI policy aligns with and complements existing IT, information security, and data privacy policies.

Data Management – Set up protocols for data collection, storage, processing, and disposal of data used by AI systems.

Define your goals

Identify your resources

Evolve constantly

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" else t
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

Emerging Risks

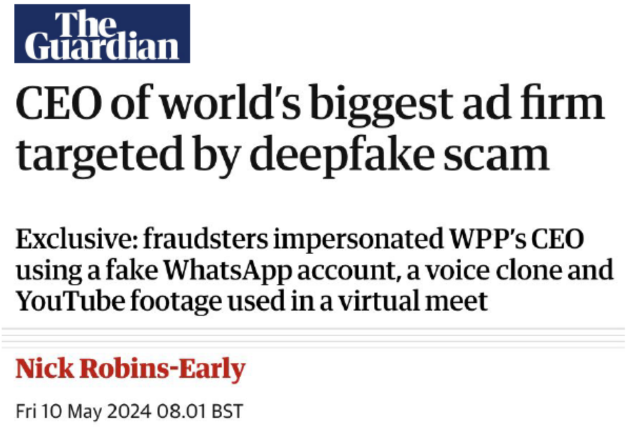
> “*I probably trust the answers that come out of ChatGPT the least of anybody on Earth*”

– Sam Altman

Founder and CEO, OpenAI

AI Risk in the Wild

Phishing & Scam



The Guardian

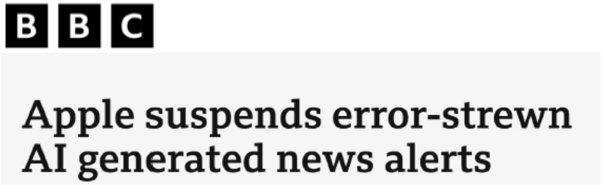
CEO of world's biggest ad firm targeted by deepfake scam

Exclusive: fraudsters impersonated WPP's CEO using a fake WhatsApp account, a voice clone and YouTube footage used in a virtual meet

Nick Robins-Early

Fri 10 May 2024 08.01 BST

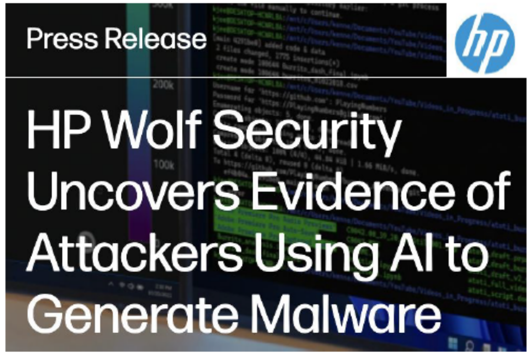
Hallucinations



BBC

Apple suspends error-strewn AI generated news alerts

Malware

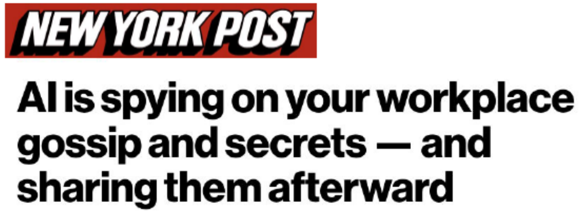


Press Release

hp

HP Wolf Security Uncovers Evidence of Attackers Using AI to Generate Malware

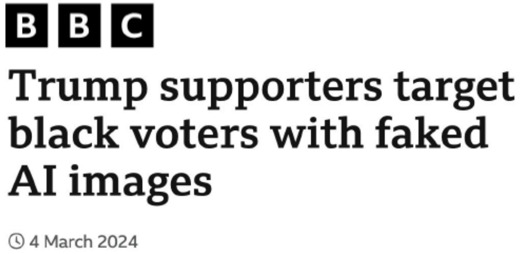
Privacy & legal issues



NEW YORK POST

AI is spying on your workplace gossip and secrets — and sharing them afterward

Disinformation

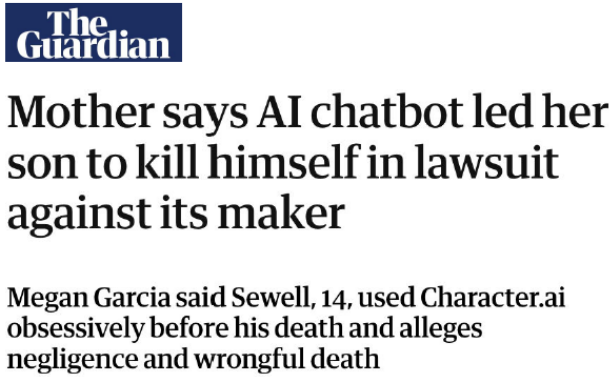


BBC

Trump supporters target black voters with faked AI images

4 March 2024

Societal issues



The Guardian

Mother says AI chatbot led her son to kill himself in lawsuit against its maker

Megan Garcia said Sewell, 14, used Character.ai obsessively before his death and alleges negligence and wrongful death

Arup Deepfake Scam

- Arup is a British multinational design and engineering company behind world-famous buildings such as the Sydney Opera House.
- Early in 2024, one of its employees in Hong Kong transferred $25 million following a video call with senior management.
- Except, it turned out, the employee hadn't been talking to Arup managers at all, but to deepfakes created by artificial intelligence.
- The employee had been tricked into sending $25 million to criminals.

> "What happened at Arup – I would call it technology-enhanced social engineering. It wasn't even a cyberattack in the purest sense. None of our systems were compromised, and there was no data affected. People were deceived into believing they were carrying out genuine transactions that resulted in money leaving the organization."
>
> **– Rob Greig**
>
> *Arup's Chief Information Officer*

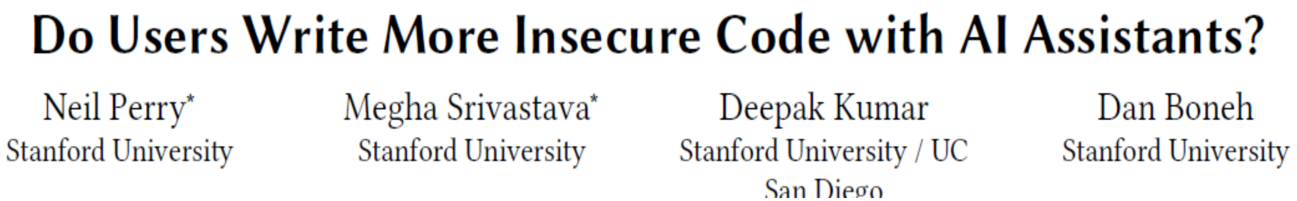
Open Worldwide Application Security Project (OWASP) LLM Top 10

1. Prompt Injection
2. Insecure Output Handling
3. Training Data Poisoning
4. Model Denial of Service
5. Supply Chain Vulnerabilities
6. Sensitive Information Disclosure
7. Insecure Plugin Design
8. Excessive Agency
9. Overreliance
10. Model Theft

Generative Coding Considerations

- Intellectual Property Ownership and Licensing Are Unsettled
- “Fair Use” in Code Training is Under Litigation
- Privacy Risks Extend Beyond Source Code
- Security Vulnerabilities Can Be Amplified by Automation
- Regulation is Shifting from Voluntary to Binding Compliance

Emerging Generative Coding Risks



Do Users Write More Insecure Code with AI Assistants?

Neil Perry* Stanford University

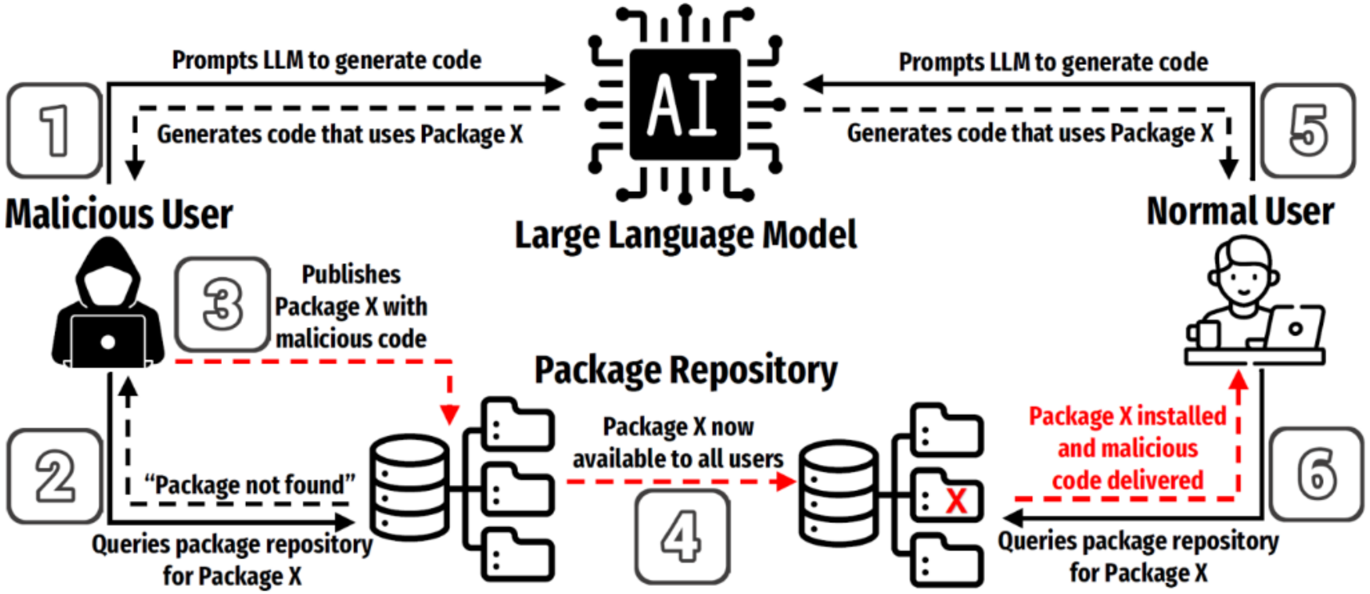
Megha Srivastava* Stanford University

Deepak Kumar Stanford University / UC San Diego

Dan Boneh Stanford University

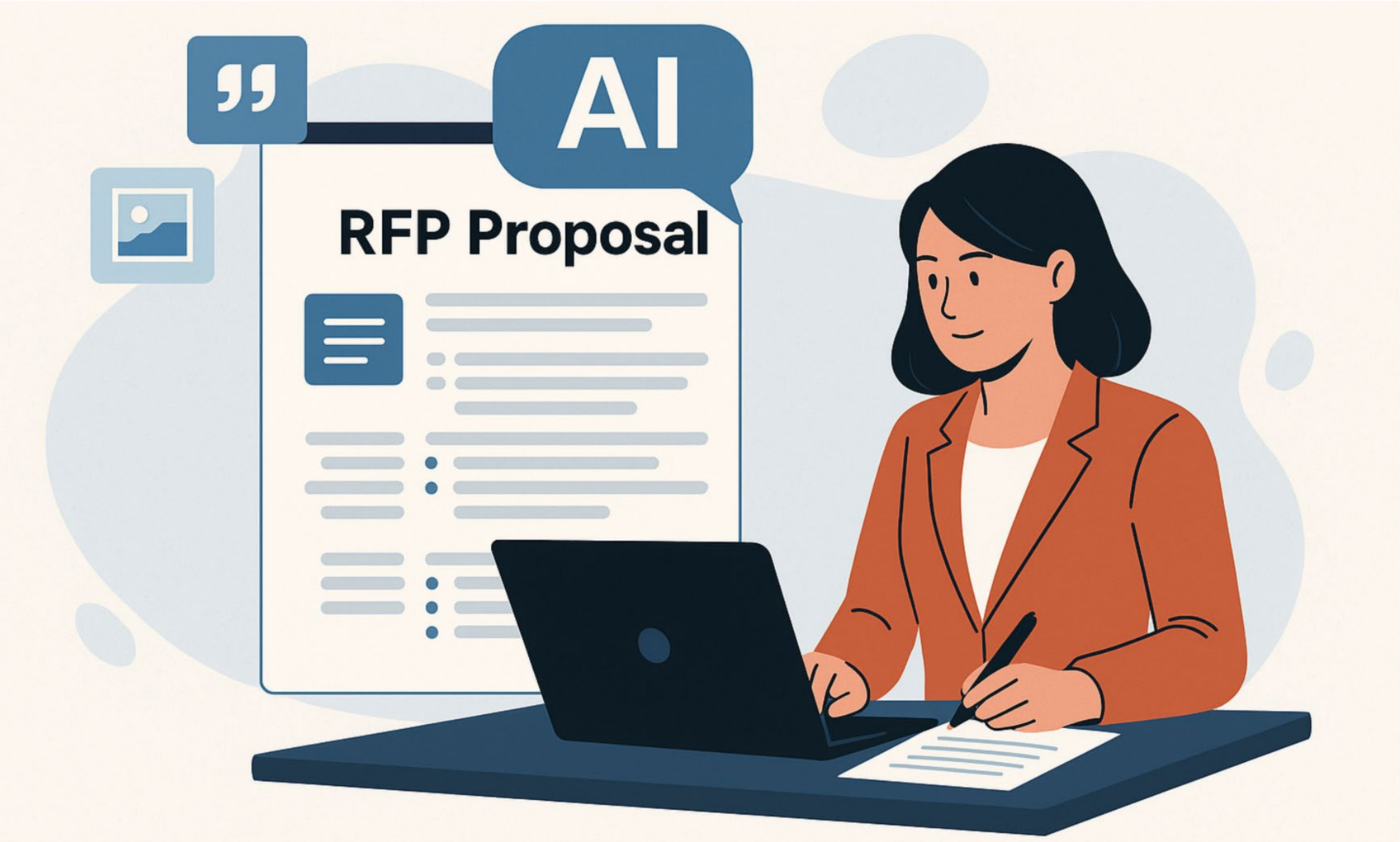
CCS '23: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, November 2023, Pages 2785-2799

We conducted the first user study examining how people interact with an AI code assistant (built with OpenAI's Codex) to solve a variety of security related tasks across different programming languages. We observed that participants who had access to the AI assistant were more likely to introduce security vulnerabilities for the majority of programming tasks, yet were also more likely to rate their insecure answers as secure compared to those in our control group. Additionally, we found that participants who invested more in the creation of their queries to the AI assistant, such as providing helper functions or adjusting the parameters, were more likely to eventually provide secure solutions.


Prompts LLM to generate code
Generates code that uses Package X
1
Malicious User
AI
Large Language Model
Prompts LLM to generate code
Generates code that uses Package X
5
Normal User
3
Publishes Package X with malicious code
Package Repository
2
"Package not found"
Queries package repository for Package X
Package X now available to all users
4
X
Package X installed and malicious code delivered
6
Queries package repository for Package X

Flowchart for Exploiting Package Hallucinations. An attacker prompts an LLM for code (1) and the generated code contains a hallucinated package name (2). The attacker publishes a package containing malicious code using the hallucinated name (3) which is now available to any user (4). Now, the next time a normal user asks a similar question to the LLM (5) and the generated code contains the same hallucinated package name, that user is at risk of installing the malware on their device (6).

Can I Use AI to Prepare an RFP Response?



Using AI to Prepare an RFP Response

Risk Considerations

- Sensitive Information Disclosure
- Data Retention and Training
- Model Reliability and Hallucination
- Supply Chain and Model Origin
- Compliance with Procurement Integrity Rules
- Overreliance on AI Outputs

Acceptable Uses with Controls

- Summarization of Publicly Available RFP Sections
- Drafting Boilerplate Text
- Grammar and Style Review of Non-Sensitive Drafts
- Research Assistance
- Outline Generation Based on Internal Templates

Prohibited Uses

- Uploading CUI, Proprietary, or Export-Controlled Data
- Submitting Proposal Content to Public LLMs
- Relying on LLM Output Without Human Review
- Using AI to Misrepresent Capabilities
- Integrating Unvetted AI Plugins or APIs
- Circumventing Company Review Processes

Implementation Safeguards

- Deploy AI in a Controlled Environment
- Data Classification Checks Before AI Use
- AI Usage Logging and Audit
- Training on AI Risks
- Clear Written Policy

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" else t
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

What is an AI Incident?

Definitions

An **AI incident** is *any event in which the behavior of an AI system – whether through error, misuse, or malicious interference – directly or indirectly causes, or could reasonably have caused, harm to people, organizations, property, fundamental rights, the environment, or the public interest*. To qualify as AI-specific harm four elements must be present: 1) a **harmed entity**, 2) a **harmful event or condition**, 3) a **causal link** to 4) an **AI system's behavior**.

EU law adds that a **"serious incident"** occurs when the AI malfunction might lead to death, serious injury, or serious damage to fundamental rights, property, safety (management or operation of critical infrastructure), or the environment.

How Does This Differ From a Cybersecurity Incident?

Dimension	Cybersecurity Incident	AI Incident
Core Focus	Confidentiality, Integrity, Availability (CIA) of information assets	Socio-technical harms spanning safety, fairness, privacy, misinformation, autonomy, and physical impacts
Primary Threat	External adversary exploiting technical vulnerabilities	Can arise from model design, data, deployment context, user misuse, or malicious attack
Failure Mode	Exploit or breach of a security control or vulnerability	Model misspecification, drift, hallucination, bias, reward hacking, adversarial attack, or security breach
Lifecycle Stage	Mostly operational / post-deployment	Pre-deployment (training data, model design), deployment, and post-deployment
Regulation	Data-protection, breach-notification, critical-infrastructure laws	Emerging AI-specific regimes plus sectoral safety, privacy, consumer protection, and antidiscrimination laws

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
  dupli_ob = pset.dupli_object
  if dupli_ob is not None and draw_dat.instances_write_dupli:
      filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
      if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
          filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
      filepath = "".join(filepath)
      dupli_world = dupli_ob.matrix_world.copy()
      transl_inv = Matrix.Translation(-dupli_world.translation)
      dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
      filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
      dupli_ob.matrix_world = dupli_world
      obj.matrix_world = Matrix.Identity(4)
      writeObject(context, instance_filepath, [obj])
      obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
  csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
  fh = open(csv_file, "w")
  for p in particles:
      rot = Quaternion.to_matrix(p.rotation).to_4x4()
      if (pset.type == "HAIR"):
          h1 = p.hair_keys[0].co
          h2 = p.hair_keys[-1].co
          loc = Matrix.Translation(h1)
          scale = Matrix.Scale((h2 - h1).length, 4)
          rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
      else:
          loc = Matrix.Translation(p.location)
          scale = Matrix.Scale(p.size, 4)
      t = loc * rot * scale
      t = emitter.matrix_world * t if pset.type == "HAIR" else t
      writeTransform(oct_t[0] * t * oct_t[1], fh)
  fh.close()
  filenames.append(csv_file)
  return filenames
except IOError as err:
  raise ExportException(msg)
```

AI Incident Classification

Why Classification Matters

Purpose

Classification aligns the team on what is harmed, why it happened, where it arose, and who owns the fix, so containment, evidence, notifications, and remediation are correct the first time.

What it unlocks

1. **Right playbook, fast:** Maps to the correct containment tactics and evidence to preserve.
2. **Clear accountability:** Names the **Incident Commander(IC)**, **Technical Owner**, **Compliance Owner**, and **Execution Team**.
3. **Accurate legal posture:** Surfaces discrimination, consumer-protection, privacy, or sector rules even without a breach.
4. **Better recovery:** Directs root-cause analysis and validation tests; prevents recurrence.
5. **Consistent metrics: Enables trend analysis across incidents and teams.**

The Lens: Domain x Failure Mode x Lifecycle x Owner

Domain (Category) — "What is harmed?"

Drives legal risk, communication posture, and customer remedies.

Failure Mode (Root Cause) — "Why did it happen?"

Selects containment steps and evidence (prompts, system prompt, tool traces, indices, model hash).

Lifecycle Stage — "Where seeded and where surfaced?"

- **Seeded:** L1 Pre-deployment; L2 Deployment / Integration; L3 Post-deployment / Operations.
- **Surfaced:** L1 / L2 / L3 where detected.

Ensures fixes land in the right phase (requirements, integration, or operations) and that gates and controls are implemented and/or strengthened.

Owner — "Who leads and who executes?"

- **Incident Commander (IC):** Legal, Security / IR, Product, or Trust & Safety (choose based on Domain).
- **Technical Owner:** ML / Data Science, Product / Platform, Security / Abuse, or UX.
- **Compliance Owner:** Privacy / Compliance, Employment Counsel, or Risk / ESG.
- **Execution Team:** ML Eng, Data Eng, Product / Platform, Security, Trust & Safety Ops, HR, Support, Communications.

AI Risk Domains

- **Discrimination & Toxicity:** Unfair treatment or harmful content targeting groups or individuals.
- **Privacy & Security:** Exposure, inference, misuse of sensitive data, or compromise of models.
- **Misinformation:** False or misleading content that distorts decisions.
- **Malicious Use & Misuse:** Intentional weaponization or use outside intended scope.
- **Human–AI Interaction:** Harm from overreliance, manipulation, or misleading UX.
- **Socioeconomic & Environmental:** Broader externalities (labor, market, resource impacts).
- **System Safety, Failures & Limitations:** Unsafe or unreliable behavior due to technical limits.

The risk domains used here are adapted from the MIT AI Risk Repository’s Domain Taxonomy

MIT AI Risk Repository https://airisk.mit.edu/

AI Failure Modes

- **Design & Data:** Flawed objectives, biased or poisoned data.
- **Model Behavior:** Hallucination, emergent harmful content, unsafe code suggestions.
  - **RAG / Grounding Failures:** Outdated or untrusted sources; retrieval misses; fabricated citations.
  - **Agent / Tool Orchestration Errors:** Unsafe tool calls; missing constraints; ambiguous goals.
- **Privacy Leakage & Inference:** Membership inference; training-data extraction; prompt-based PII disclosure.
- **Security Exploits & Supply-Chain:** Prompt injection; model theft; third-party compromise.
- **Operational Drift:** Concept drift, feedback-loop amplification.
- **Malicious Use:** Deep-fake fraud and extortion, automated disinformation campaigns, emotional manipulation.
- **Human–AI Interaction & Overreliance:** Automation bias; missing escalation paths.
- **System Safety & Limitations:** Reward hacking; evaluation blind spots; unsafe constraints.

Lifecycle and Owners

Lifecycle (Seeded / Surfaced)

- **Pre-deployment:** Problem framing, policy, data, training, evaluation design.
- **Deployment / Integration:** RAG sources, tool permissions, access controls, UX.
- **Post-deployment / Operations:** Real-world use, drift, abuse, monitoring, incident handling.

Owners

- **Incident Commander (IC):** Managing incident response. **(Domain)**
- **Primary Technical Owner:** Identifies failure mode and fixes the system. **(Failure Mode)**
- **Policy / Compliance Owner:** Determines compliance obligations. **(Domain)**
- **Execution Teams:** Support Incident Response. **(Lifecycle)**

AI Incident Response Essentials

1. **Identify & Classify:** Confirm AI involvement; tag Domain–Failure Mode–Lifecycle–Owner; assign Incident Commander; set initial severity.
2. **Contain & Stabilize (Safety-first):** Stop harmful behavior: safe-mode / human-in-the-loop, rate limits, disable tool calls, geofence features, bind RAG to trusted sources.
3. **Preserve Evidence & Notify:** Legal hold; capture prompts, outputs, logs, model / version, configs, retrieval sources, chain-of-custody; determine internal and regulatory notifications.
4. **Assess Impact & Risk:** Identify harmed entities, scope, and business / legal exposure; apply your AI-harm test; decide on customer support and comms posture.
5. **Remediate & Recover:** Fix root cause, patch guardrails, retune or retrain; canary rollout with evals; vendor coordination.
6. **Document & Learn:** Conduct a blameless post-incident review; update playbooks, metrics, dashboards; add red-team tests to prevent recurrence.

Real World Examples

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" else t
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

Chatbots Gone Wrong

Microsoft Tay (2016)

What happened: Bot produced racist / offensive content; offline within 16 hours.

Failure mode: Model behavior + malicious prompts; guardrail gaps.

MIT Harm: Discrimination & Toxicity.

Fix: Abuse hardening, identity & rate limits, supervised safety filters.

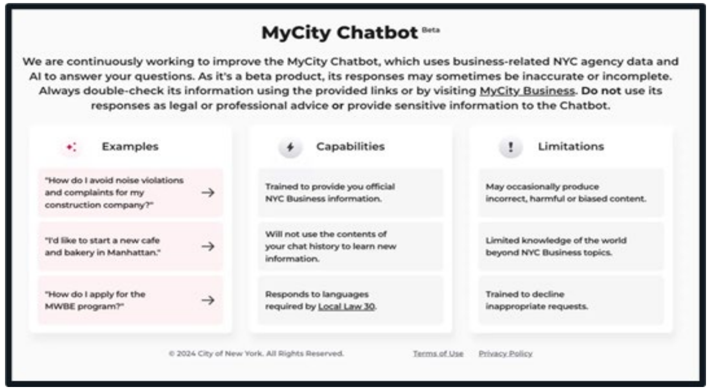
Air Canada Chatbot (2022 - 2024)

What happened: Chatbot promised bereavement discount contrary to policy; tribunal held airline liable.

Failure mode: RAG / grounding misinformation; policy misalignment.

MIT Harm: Misinformation; Human–AI Interaction.

Fix: Bind to authoritative policy, verify citations, require legal approval on policy claims.



MyCity Chatbot Beta

We are continuously working to improve the MyCity Chatbot, which uses business-related NYC agency data and AI to answer your questions. As it's a beta product, its responses may sometimes be inaccurate or incomplete. Always double-check its information using the provided links or by visiting MyCity Business. **Do not** use its responses as legal or professional advice or provide sensitive information to the Chatbot.

Examples	Capabilities	Limitations
"How do I avoid noise violations and complaints for my construction company?" →	Trained to provide you official NYC Business information.	May occasionally produce incorrect, harmful or biased content.
"I'd like to start a new cafe and bakery in Manhattan." →	Will not use the contents of your chat history to learn new information.	Limited knowledge of the world beyond NYC Business topics.
"How do I apply for the MWBE program?" →	Responds to languages required by Local Law 30.	Trained to decline inappropriate requests.

© 2024 City of New York. All Rights Reserved. Terms of Use Privacy Policy

NYC MyCity (2023 - 2024)

What happened: Bot advised actions that would violate housing and labor law.

Failure mode: RAG / grounding failures on regulated topics.

MIT Harm: Misinformation; Human–AI Interaction.

Fix: Curated corpus, refusal & human escalation for legal queries, continuous evaluations.

Biased and Unfair AI Behavior

iTutorGroup (2020)	RiteAid (2012-2020)
What happened: Application auto-rejected female applicants 55+ and male applicants 60+ for online tutor roles. **Failure mode:** Design & Data (mis-specified objective and screening rules); System Safety & Limitations (unchecked automated filters). **MIT Harm domain:** Discrimination & Toxicity.	**What happened:** Deployed facial recognition in hundreds of stores, which misidentified customers (with elevated risks for people of color and women), leading to wrongful stops. **Failure mode:** Security Exploits & Supply-Chain (vendor risk, accuracy not validated); Privacy Leakage & Inference; Human–AI Interaction. **MIT Harm domain:** Privacy & Security; Discrimination & Toxicity; Human–AI Interaction.

It is Not Hype but …it is Not Ready

McDonalds (2024)	Zillow (2021)
What happened: After ~2.5 years of testing an AI Automated Order Taker, McDonald's decided to end the pilot by July 26, 2024. Reports cited order-accuracy issues and viral videos showing misrecognitions. **Failure mode:** Model Behavior; Human–AI Interaction (low-confidence outputs with no graceful handoff); System Safety & Limitations (accuracy below operational threshold). **MIT Harm domain:** Human–AI Interaction; AI System Safety, Failures & Limitations.	**What happened:** Zillow announced it would wind down Zillow Offers and lay off ~2,000 employees (~25%), citing unpredictability in forecasting home prices and losses. **Failure mode:** System Safety & Limitations (forecast precision below business threshold); Design & Data (objective / metric mismatch); Operational Drift (pandemic-era volatility). **MIT Harm domain:** AI System Safety, Failures & Limitations; Socioeconomic & Environmental (market disruption, layoffs).

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" else t
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

Recommendations

Staying Safe From Deepfakes

- Phishing-resistant MFA
- Live verification (audio / video) to detect anomalies
- Use of deepfake detection tools and open-source validation (e.g., image metadata analysis)
- Enhanced due diligence for suspicious patterns
- Adopt risk appropriate controls

Generative Coding Best Practices

- Maintain Human Oversight and Review
- Track Data Provenance and Licensing
- Embed Privacy-by-Design in Development
- Implement Secure Coding and Testing Pipelines
- Document AI Use and Decision-Making
- Use the Proper Tools for the Job
- Establish AI Usage Guidelines and Training

For the Humans in the Loop

Employee Education and Training. Educate your workforce on the organization's AI policy and how they can and should use AI responsibly. Help them understand the benefits and limitations of AI. Promote trust and collaboration.

Accountability and Liability. Address accountability in the event of errors or biases arising from AI systems. Determine who is responsible for AI-related decisions and potential negative consequences.

Responsible AI Use. Establish clear guidelines for ethical and responsible AI use. Address issues such as bias mitigation, fairness in decision-making, and transparency in AI outputs.

Governance and Oversight. Establish a clear governance framework for AI implementation, such as a committee or team responsible for overseeing AI development, deployment, and monitoring.

FOR THE HUMANS IN THE LOOP
Employee Education and Training
Educate your organizations A/d policy and hovt they can use All responsibily. Help them understand the benefits and limitations of AI.
Accountability and Liability
Address accreentablitility in the event of errors of biases arising from AI systems.
Responsible AI Use
Establish clear guidhines for ethical and responsible
Governance and Oversight
Establish a clear governance framework for AI implemorotion, such as a committee or team responsible for oversening AI development, deployment, and monitoring.
PREPARED
PROFICIEN
PROACTIVE

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" el
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

Key Issues to Consider When Procuring AI Tools

Evolving Gen AI Terms of Use

- **Broad Disclaimers of Warranty and Liability, both for:**
  - The **AI Tools**
  - The **AI Outputs**
- **Customers Use at Their Own Risk.** Solely responsible for content you input and use of outputs.
- **Very Little Control Over Subsequent Use of Your Inputs.**
  - Can you opt out?
- **Broad Indemnification Obligations on Customer.**
  - Conversely, almost no provider indemnification of customer.

Best Practices for Contracting for AI Services

- **Define Scope and Permitted Uses Clearly.** Specify intended use cases, prohibited uses, and any regulatory restrictions (e.g., no deployment in high-risk AI contexts without approval).
- **Address IP Ownership and Training Rights.** Clarify ownership of outputs, rights to underlying models, and whether your data may be used to train the provider's AI.
- **Mandate Data Governance and Privacy Compliance.** Require adherence to applicable laws (GDPR, CCPA) and include obligations for de-identification, security controls, and breach notification.
- **Set Performance, Accuracy, and Bias Standards.** Include measurable service levels, accuracy thresholds, and obligations to mitigate bias, aligned with frameworks like the EU AI Act or NIST AI RMF.
- **Require Transparency, Audit, and Termination Rights.** Ensure access to documentation, testing results, and audit rights; provide termination rights if legal or compliance risks arise.

```
    csv_path = bpath.abspath(draw_dat.csv_export_path)
pset = psys.settings
particles = [p for p in psys.particles] if pset.type == 'ALIVE']
filenames = []
if pset.render_type == "OBJECT":
    dupli_ob = pset.dupli_object
    if dupli_ob is not None and draw_dat.instances_write_dupli:
        filepath = [bpath.abspath(draw_dat.path), dupli_ob.name]
        if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
            filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
        filepath = "".join(filepath)
        dupli_world = dupli_ob.matrix_world.copy()
        transl_inv = Matrix.Translation(-dupli_world.translation)
        dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
        filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
        dupli_ob.matrix_world = dupli_world
        obj.matrix_world = Matrix.Identity(4)
        writeObject(context, instance_filepath, [obj])
        obj.matrix_world = obj_world
else:
    WARNING("Invalid datatype '%s'" % pset.render_type)
    return
try:
    csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
    fh = open(csv_file, "w")
    for p in particles:
        rot = Quaternion.to_matrix(p.rotation).to_4x4()
        if (pset.type == "HAIR"):
            h1 = p.hair_keys[0].co
            h2 = p.hair_keys[-1].co
            loc = Matrix.Translation(h1)
            scale = Matrix.Scale((h2 - h1).length, 4)
            rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
        else:
            loc = Matrix.Translation(p.location)
            scale = Matrix.Scale(p.size, 4)
        t = loc * rot * scale
        t = emitter.matrix_world * t if pset.type == "HAIR" else t
        writeTransform(oct_t[0] * t * oct_t[1], fh)
    fh.close()
    filenames.append(csv_file)
    return filenames
except IOError as err:
    raise ExportException(msg)
```

Practical Risk Mitigation

Case Study #1
Internal Use of Third-Party Gen AI

Spacely Space Sprockets, a leading producer of sprockets for space-age machinery, is considering adopting a third-party Gen AI tool for internal use by its employees. The company assumes many of its employees are using publicly-available Gen AI tools but does not currently regulate such use. What are some strategic measures that may help the company mitigate risk?

Practical Risk Mitigation Measures:

- ✓ Internal AI Policy and Training
- ✓ Selecting the Right Gen AI Tool
- ✓ Negotiation Adequate Protections in Gen AI Terms of Use

Case Study #2
Offering a Gen AI Product

ACME Corporation is planning on offering a generative AI tool, **ACME Super Mail™**, that drafts, schedules, and sends internal and customer e-mails.

Which safeguards should ACME implement to manage the related legal, security, and compliance risks?

Practical Risk Mitigation Measures:

- ✓ Governance & Oversight
- ✓ Technical Controls
- ✓ Legal & Contractual Safeguard
- ✓ Human-in-the-Loop & User Controls
- ✓ Transparency & Documentation
- ✓ Ongoing Monitoring & Audit

Questions?

Stay Connected

Michael Breslin

Partner | Atlanta

mbreslin@ktslaw.com

Meghan Farmer

Partner | Atlanta

mfarmer@ktslaw.com

Greg Silberman

Counsel | Silicon Valley

gsilberman@ktslaw.com

Appendix

Appendix

- MIT AI Risk Repository https://airisk.mit.edu/
- AI Incident Database https://incidentdatabase.ai/
- NIST AI Risk Management Framework https://www.nist.gov/itl/ai-risk-management-framework
- OWASP GenAI Security Project https://genai.owasp.org/
- AI Hallucination Cases https://www.damiencharlotin.com/hallucinations/
- Do Users Write More Insecure Code with AI Assistants https://arxiv.org/abs/2211.03622
- We Have a Package for You! A Comprehensive Analysis of Package Hallucinations by Code Generating LLMs https://www.usenix.org/conference/usenixsecurity25/presentation/spracklen
- Chihuahua or Muffin https://www.karenzack.com/work/recognition-series

Contracting Exemplars: Broad Disclaimers of Warranty / Liability

AI Tools, Damages

- ❑ ***AI Tools:*** *AI TOOLS ARE [6] PROVIDED TO CUSTOMER "AS IS" WITHOUT ANY INDEMNITIES, REPRESENTATIONS, OR WARRANTIES OF ANY KIND (WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE), AND [7] MAY BE MODIFIED, DISCONTINUED, OR CANCELLED AT ANY TIME FOR ANY REASON AT THE SOLE DISCRETION OF AI PROVIDER.*

- ❑ ***Damages:*** *IN NO EVENT WILL AI PROVIDER BE [8] LIABLE FOR ANY DIRECT, SPECIAL, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES ARISING UNDER OR IN CONNECTION WITH AI TOOLS OR CUSTOMER'S USE THEREOF.*

Contracting Exemplars: Broad Disclaimers of Warranty / Liability (cont’d)

AI Outputs

❑ ***Infringing Outputs:*** *Provider disclaims all warranties, express or implied, regarding the Output, including any implied warranties* ***that the Output will not violate the rights of a third party or any applicable law***. *You are solely responsible for the creation and use of the Output.*

❑ ***Hallucinations & Accuracy:*** *YOU ACKNOWLEDGE THAT OUTPUT IS GENERATED BY MACHINE LEARNING CAPABILITY, AND WE MAKE NO WARRANTY OR GUARANTEE AS TO THE ACCURACY, COMPLETENESS OR RELIABILITY OF THE OUTPUT. PROVIDER WILL HAVE NO LIABILITY ARISING FROM YOUR USE OF THE AI FEATURES OR ANY ERRORS OR OMISSIONS CONTAINED IN THE OUTPUTS. THE OUTPUT MAY NOT BE UNIQUE AND MAY NOT BE PROTECTABLE BY INTELLECTUAL PROPERTY RIGHTS.*

Contracting Exemplars: Customer Solely Responsible for Inputs and Outputs

Sample Excerpts:

- ❑ ***You are solely responsible for AI Content****, including the accuracy, quality, appropriateness, and legality thereof, and will ensure that your AI Content and use of AI Products does not (i) violate any applicable law; (ii) violate these AI Terms or the Agreement; or (iii) infringe, violate, or misappropriate the rights of AI Provider or any third party. AI Content means any text you type or images, content, or data you upload into AI Products ("Input"), as well as any text, images, or content generated by AI Products through your use of AI Products or through prompts you provide to AI Products ("Output", together "AI Content").*
- ❑ ***You agree that you will not include any sensitive personal data*** *of any individual (including data that reveals racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, health data or data concerning your sex life or sexual orientation) in any Input to AI Products.*
- ❑ *When using Outputs,* ***you agree to inform viewers of those Outputs that the content is AI-generated****.*

Contracting Exemplars: No Ownership or Control Over Customer Content

Sample Excerpt: Provider may use customer content to train AI tools / services

- ❑ ***Our Use of Customer Content.*** *We may use Customer Content to provide, maintain, develop, and improve our Services, comply with applicable law, enforce our terms and policies, and keep our Services safe.*

- ❑ ***Opt Out.*** *If you do not want us to use your Customer Content to train our models, you can opt out by following the instructions at this Link. Please note that in some cases this may limit the ability of our Services to better address your specific use case.*

Contracting Exemplars: Broad Customer Indemnity Obligations

- Most Gen AI providers do not indemnify end users for claims arising from AI tools or Customer’s use thereof
- Limited IP infringement exception for enterprise or business accounts (excluding end user prompts)
- Conversely, broad indemnity from end user for claims arising from:
  - Use of the AI tool and AI outputs
  - Violation of Terms of Use or Provider policies
  - Violation of third-party rights by end user prompts
  - Violation of any laws in connection with end user data or use of AI tool

Some Progress on Indemnification of End Users: Evolving “Customer Copyright Commitments”

Sample Excerpt:

(i) **Generated Output**. Google’s indemnification obligations under the Agreement also apply to allegations that an unmodified Generated Output from a Generative AI Indemnified Service using only Google Pre-Trained Model(s), a Fine-Tuned Model, or a Customer Adapter Model used with a Google Pre-Trained Model infringes a third party’s Intellectual Property Rights.

(i) **Exceptions:** This subsection (i) (Generated Output) does not apply if the allegation relates to a Generated Output where: (1) Customer creates or uses such Generated Output **that it knew or should have known was likely infringing**, (2) Customer (or Google at Customer’s instruction) **disregards, disables, or circumvents source citations, filters, instructions, or other tools Google makes available to help Customer create or use Generated Output responsibly**, (3) Customer uses such Generated Output **after receiving notice of an infringement claim** from the rightsholder or its authorized agent, (4) the allegation is based on **a trademark-related right as a result of Customer’s use of such Generated Output in trade or commerce**, or (5) Customer **does not have the necessary rights to the Customer Data used to customize or retrain** the Fine-Tuned Model or Customer Adapter Model. “Generative AI Indemnified Service” means a Service or feature listed at https://cloud.google.com/terms/generative-ai-indemnified-services, **where the use of such Service or feature is paid for by Customer and not subject to credits or free tier usage**.

(ii) **Training Data**. Google’s indemnification obligations under the Agreement also apply to allegations that Google’s use of training data to create any Google Pre-Trained Model utilized by a Generative AI Service infringes a third party’s Intellectual Property Rights. This indemnity does not cover allegations related to a specific Generated Output, which may be covered by subsection (i) (Generated Output) above.

AI Contracting Practice Pointers: A Few Do’s & Don’ts

- ✓ Thoroughly vet Gen AI tool prior to use
- ✓ Deploy gen AI tool on protected enterprise platform where possible
- ✓ Implement controls on data inputs
- ✓ Prohibit commingling inputs with AI training data
- ✓ Factor in ownership limits in AI outputs
- ✓ Verify quality and integrity of outputs
- ✓ Examine output for possible IP infringement
- ✓ Keep records of AI tool used to generate any content to be used in company products or services
- ✓ Protect company by being proactive. Ensure settings do not permit the sharing of any unnecessary data or information
- ✓ Establish a process to review and update Gen AI use policy
- ✓ Continually monitor emerging uses cases and laws

- X Do not assume accuracy or confidentiality of Gen AI tool
- X Do not disclose customer or employee data, IP, trade secrets or information deemed “Internal Use” or “Confidential” except on a protected platform approved for that purpose
- X Do not use open-source AI tool in customer-facing applications
- X Do not use public datasets for training AI models unless approved
- X Do not enable Gen AI in business systems until an appropriate evaluation has been completed
- X Do not fail to negotiate adequate protections in your vendor contracts involving Gen AI functionality

Kilpatrick

Kilpatrick Confidential

ktslaw.com

Anchorage | Atlanta | Augusta | Beijing | Charlotte | Chicago | Dallas | Denver | Houston | Los Angeles New
York | Phoenix | Raleigh | San Diego | San Francisco | Seattle | Shanghai | Silicon Valley
Stockholm | Tokyo | Walnut Creek | Washington D.C. | Winston-Salem