Contestable AI by Design

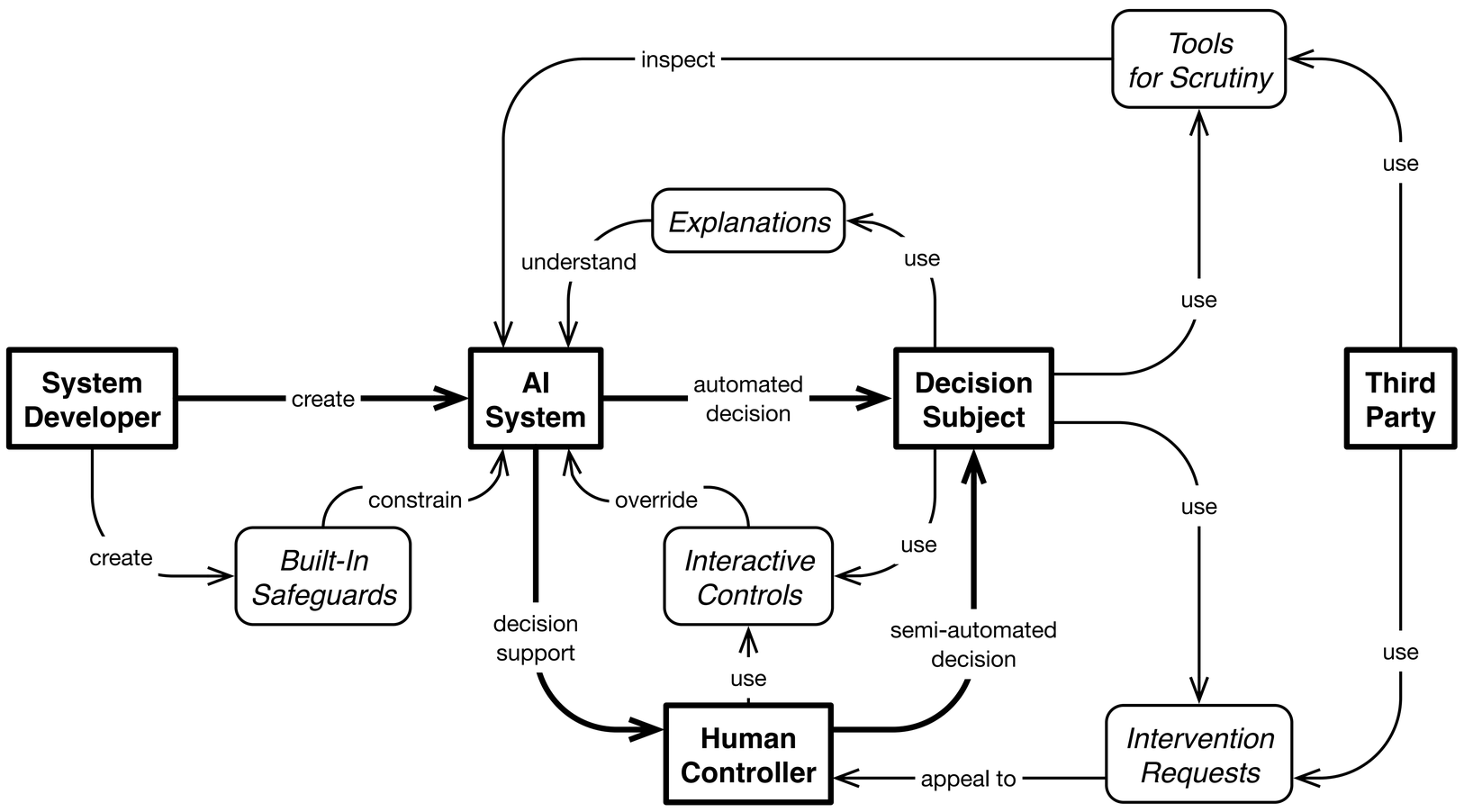
To ensure artificial intelligence systems respect people's autonomy, they must be **contestable**.

Contestable AI systems are:
– open and responsive to human intervention,
– throughout their lifecycle,
– emphasizing a dialogical relationship with decision subjects.

Read the article:
Alfrink, K., Keller, I., Kortuem, G., & Doorn, N. (2022). Contestable AI by Design: Towards a Framework. *Minds and Machines*. https://doi.org/10/gqnjcs

This framework describes **five system features** and **six development practices** contributing to AI system contestability mapped to typical human-AI system actors and system development lifecycle phases.

Features

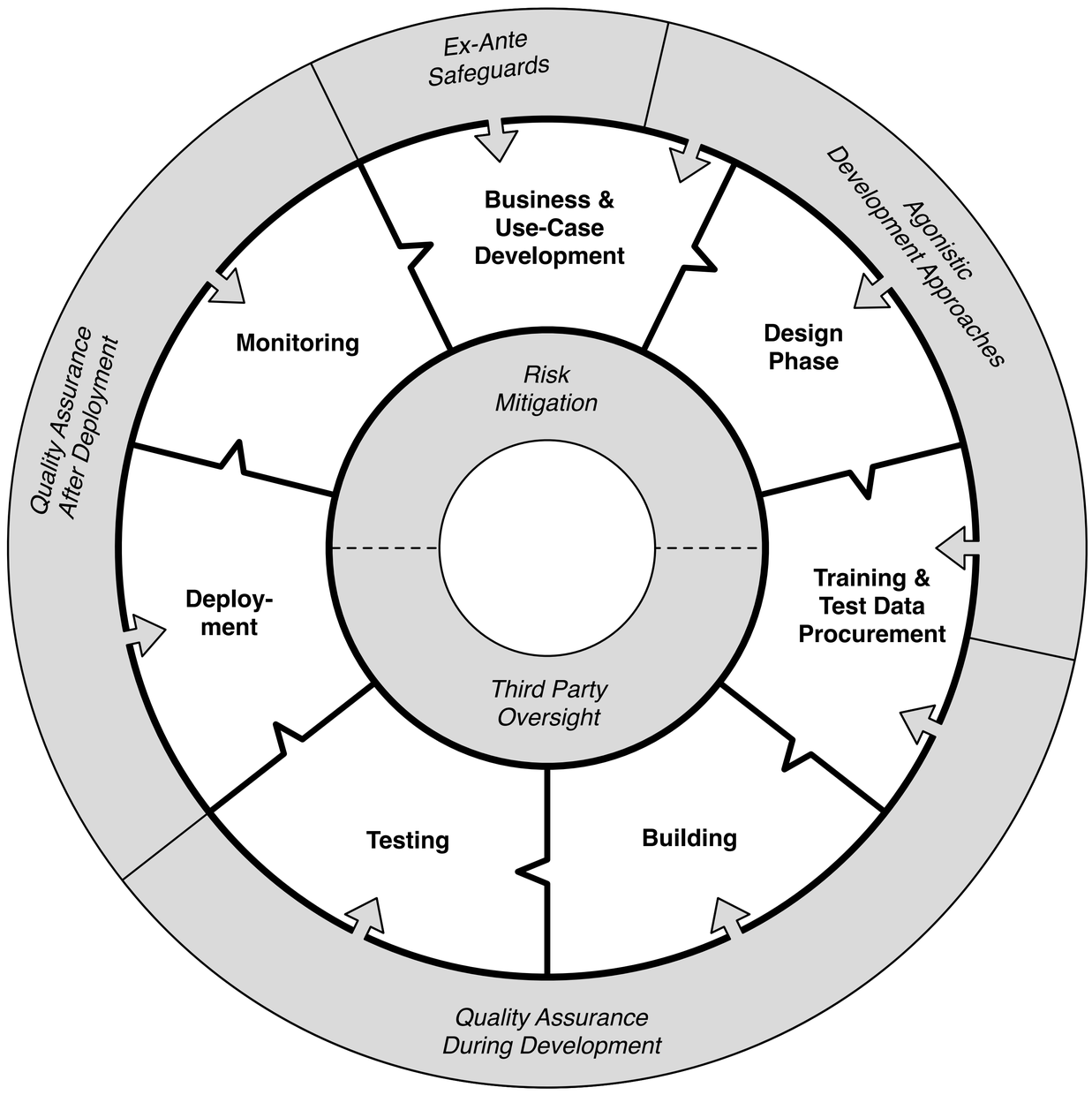


System developers create *built-in safeguards* to constrain the behavior of AI systems. **Human controllers** use *interactive controls* to correct or override AI system decisions. **Decision subjects** use *interactive controls*, *explanations*, *intervention requests*, and *tools for scrutiny* to contest AI system decisions. **Third parties** also use *tools for scrutiny* and *intervention requests* for oversight and contestation on behalf of individuals and groups.

FEATURE	EXAMPLES
Built-In Safeguards	External adversarial system; formal constraints.
Interactive controls	Negotiate, correct, or override machine decisions; feedback loop back to training; supplement local contextual data.
Explanations	Traceable decision chains; behavioral explanations; sandboxing; local approximations; justifications.
Intervention Requests	Human review; supportive, synchronous channels; third-party representation; collective action; dialectical exchange.
Tools for Scrutiny	Norms linked to implementation; documentation; formal proofs; comparative measures; opaque assurances.

Practices

During **business and use-case development**, *ex-ante safeguards* are put in place to protect against potential harms. During **design** and **procurement of training and test data**, *agonistic development approaches* enable stakeholder participation, making room for and leveraging conflict towards continuous improvement. During **building** and **testing**, *quality assurance* measures are used to ensure stakeholder interests are centered and progress towards shared goals is tracked. During **deployment** and **monitoring**, further *quality assurance* measures ensure system performance is tracked on an ongoing basis, and the feedback loop with future system development is closed. Finally, throughout, *risk mitigation* intervenes in the system context to reduce the odds of failure, and *third party oversight* strengthens the role of external reviewers to enable ongoing outside scrutiny.



FEATURE	EXAMPLES
Ex-Ante Safeguards	Anticipating impacts; acceptance criteria; certification.
Agonistic Dev Approaches	Co-construct decision-making process; ongoing adversarial dialogue.
QA Measures During Dev	Stakeholder needs guiding development; bias prevention; living labs; stakeholder feedback.
QA Measures After Deploy	Procedural integrity; monitoring for bias & misuse; feedback from corrections, appeals & additional contextual info.
Risk Mitigation	User education; environmental limits
Third-Party Oversight	Model-centric tools for auditing; trusted intermediaries; secure environments.