Constraint-Driven Deployment on the Cloud

Ouail Derghal
ouail.derghal@imt-atlantique.fr

P4S, IMT Atlantique, Lab-STICC

P4S Monthly Meeting, March 2025

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Lab-STICC

→ Problem:
- Traditional deployment processes are manual, error-prone, and rigid
- Security and infrastructure constraints are often considered late in the process

→ Objective of this talk:
- Introduce a deployment process that starts with constraints and requirements
- Show how deployments can be automatically computed and executed

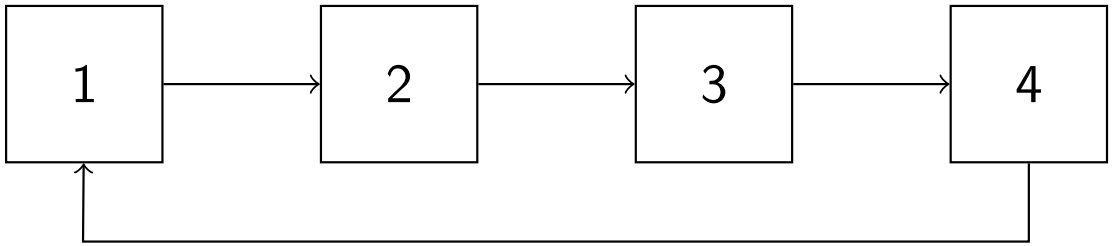
→ Overview:

- Developers manually define deployment steps
- Infrastructure and security constraints are often handled separately
- DevOps teams write and maintain deployment scripts manually
- Changes require manual intervention and can introduce inconsistencies

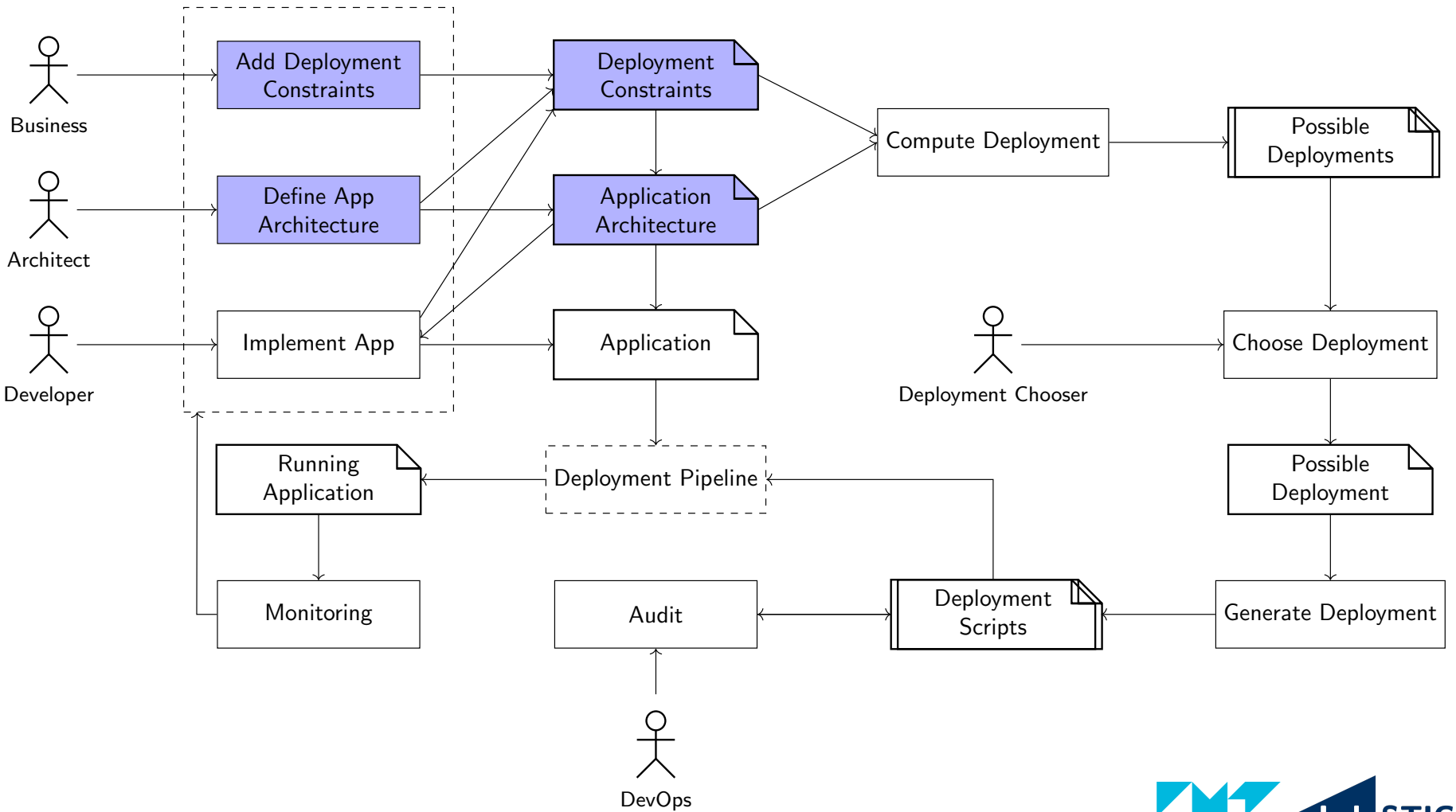
→ Challenges:

- Increased complexity of deployment as applications grow
- Technical expertise required, making maintenance challenging for non-experts
- Risk of vendor lock-in, limiting flexibility
- Challenges in multi-cloud environments, including interoperability and consistency across platforms

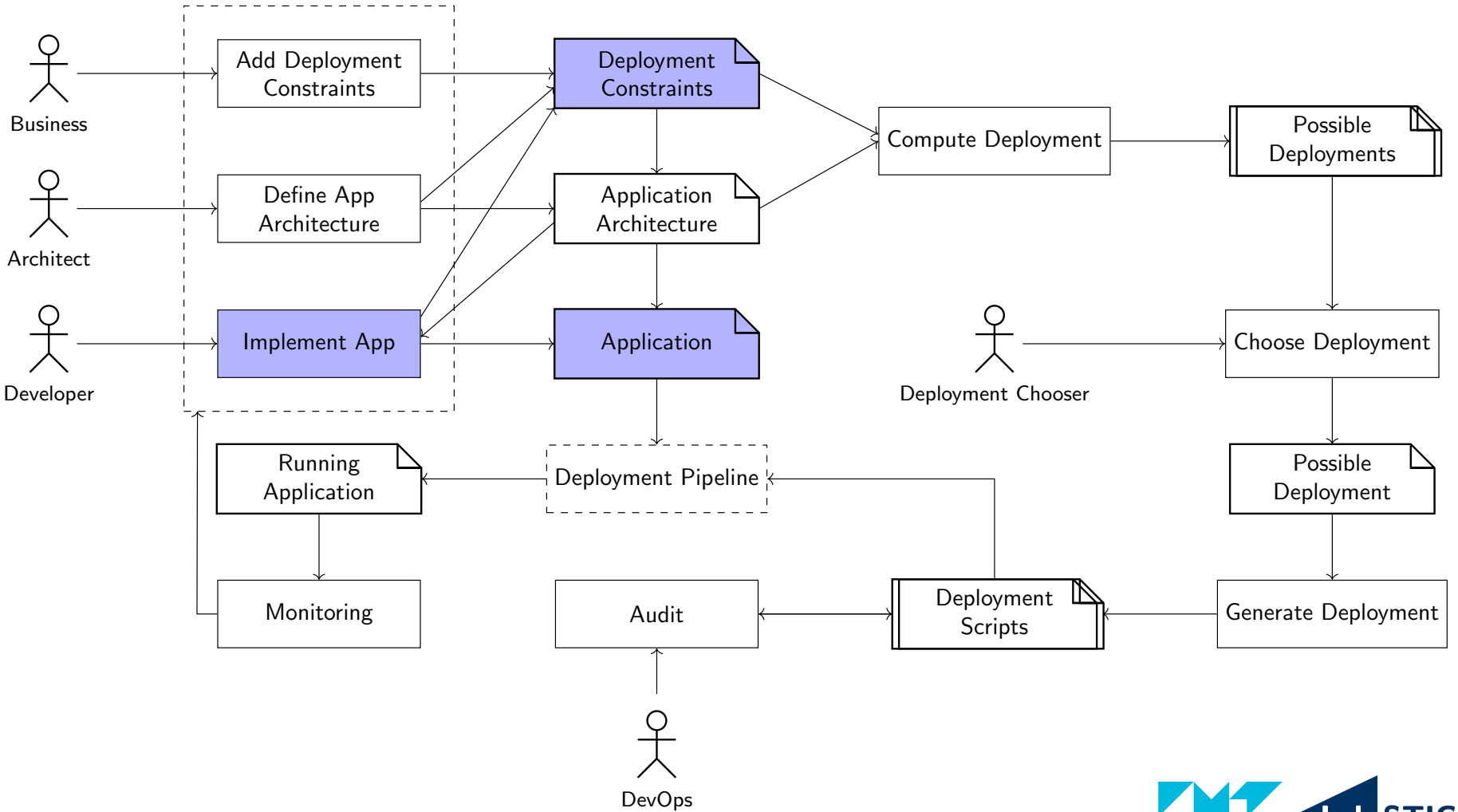
- *Key idea:* a deployment is **computed** based on constraints
- A valid deployment is choosed from the list of feasible deployments
- Deployment scripts are generated from the chosen deployment
- The deployment process flow:
  1. Architecture and constraints definition
  2. Deployment planning and feasibility
  3. Deployment selection and script generation
  4. Application deployment and monitoring



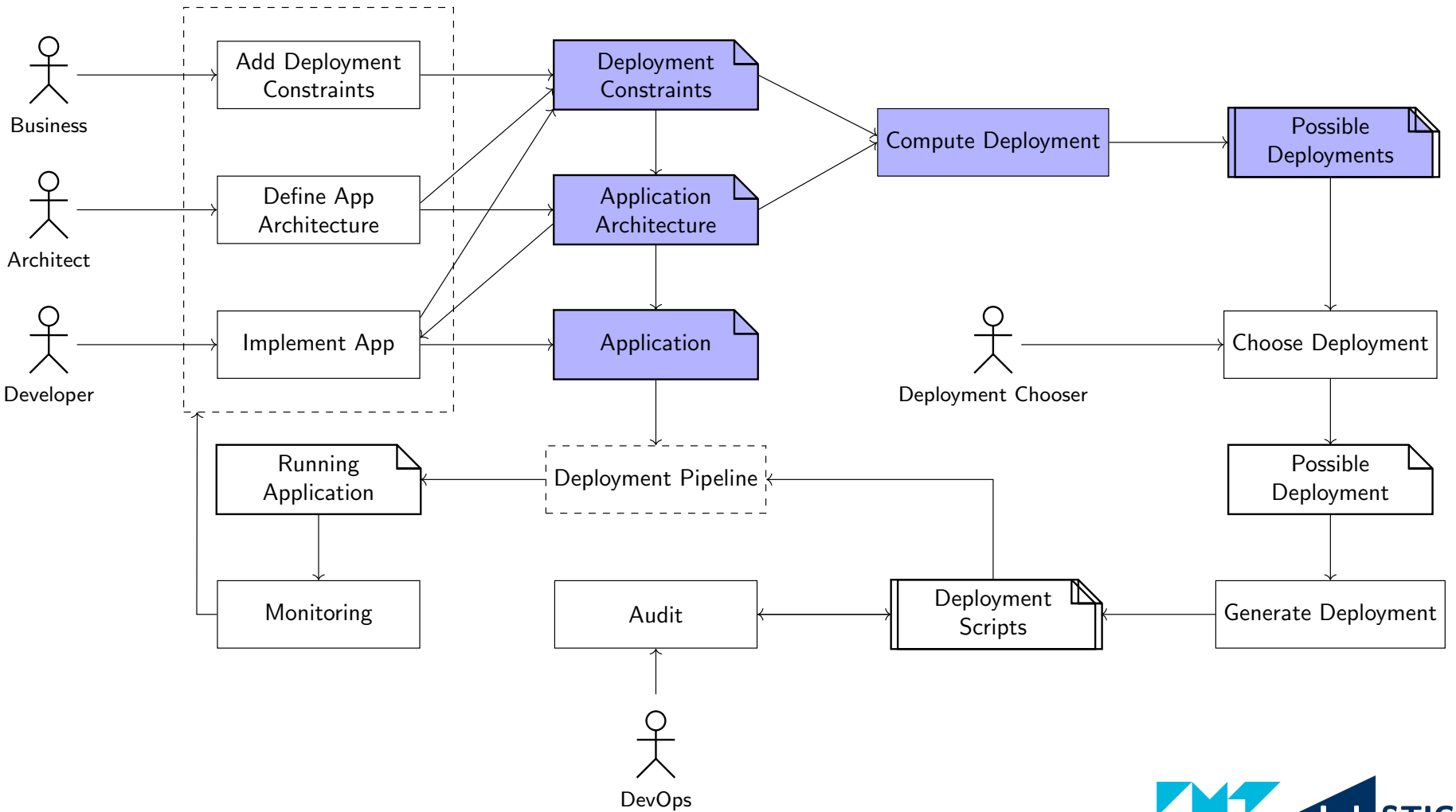
- **Goal:** Specify the application's architecture and deployment constraints
- **Who is involved?** Architects, Business Actors
- **Key Inputs:**
  → Security constraints
  → Infrastructure constraints
- **Results:**
  → Application architecture (components, services and dependencies)
  → Set of constraints that guide deployment decisions


Business
Architect
Developer
Add Deployment Constraints
Define App Architecture
Implement App
Deployment Constraints
Application Architecture
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Possible Deployment
Generate Deployment
Deployment Scripts
Audit
DevOps
Deployment Pipeline
Running Application
Monitoring

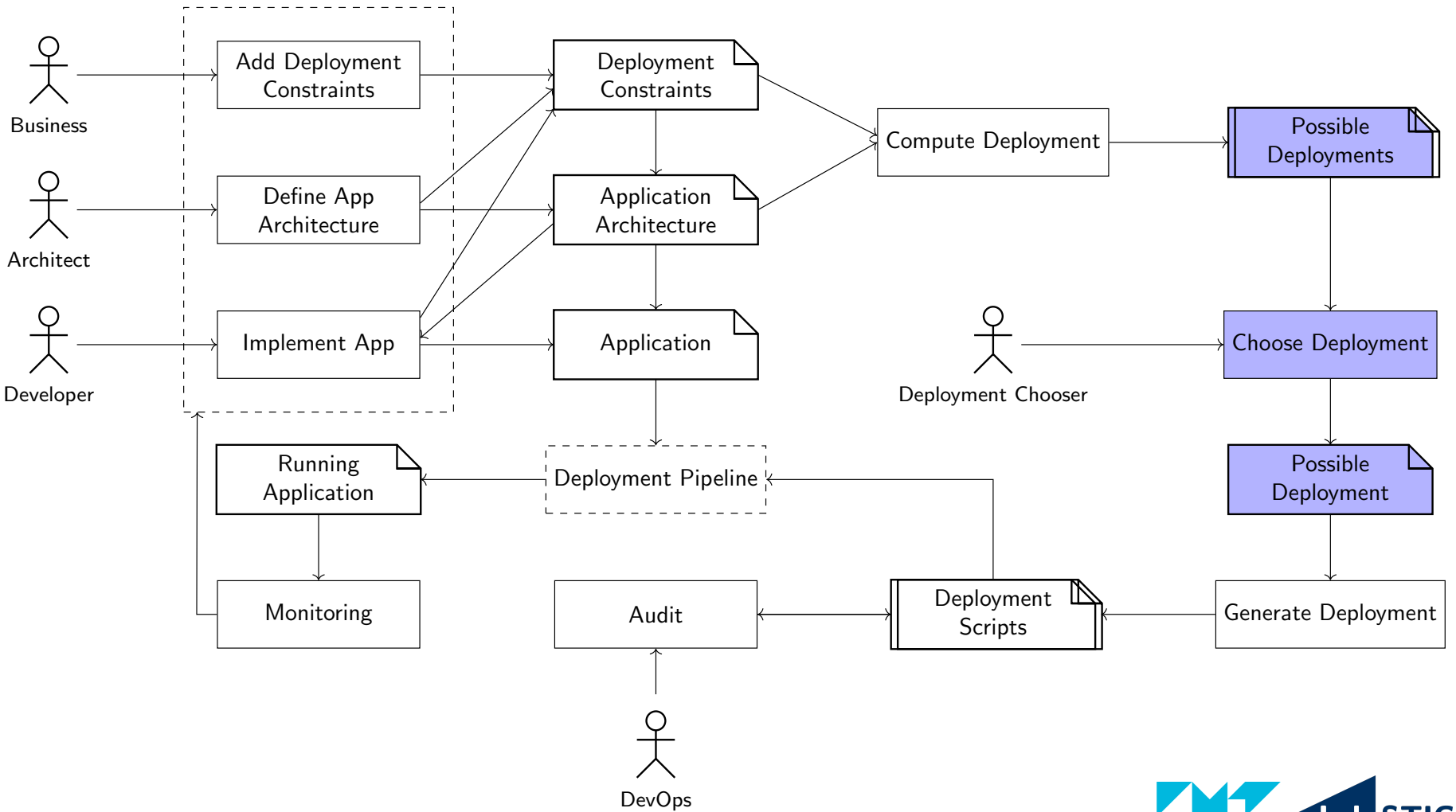
- **Goal:** Develop and package the application based on the architecture
- **Who is involved?** Developers
- **Key Activities:**
  → Code implementation
  → Integration of necessary libraries and services
  → Definition of deployment constraints
- **Results:**
  → Deployable application artifacts (container images, binaries, ...)


Business
Architect
Developer
Add Deployment Constraints
Define App Architecture
Implement App
Deployment Constraints
Application Architecture
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Possible Deployment
Generate Deployment
Deployment Scripts
Audit
DevOps
Deployment Pipeline
Running Application
Monitoring

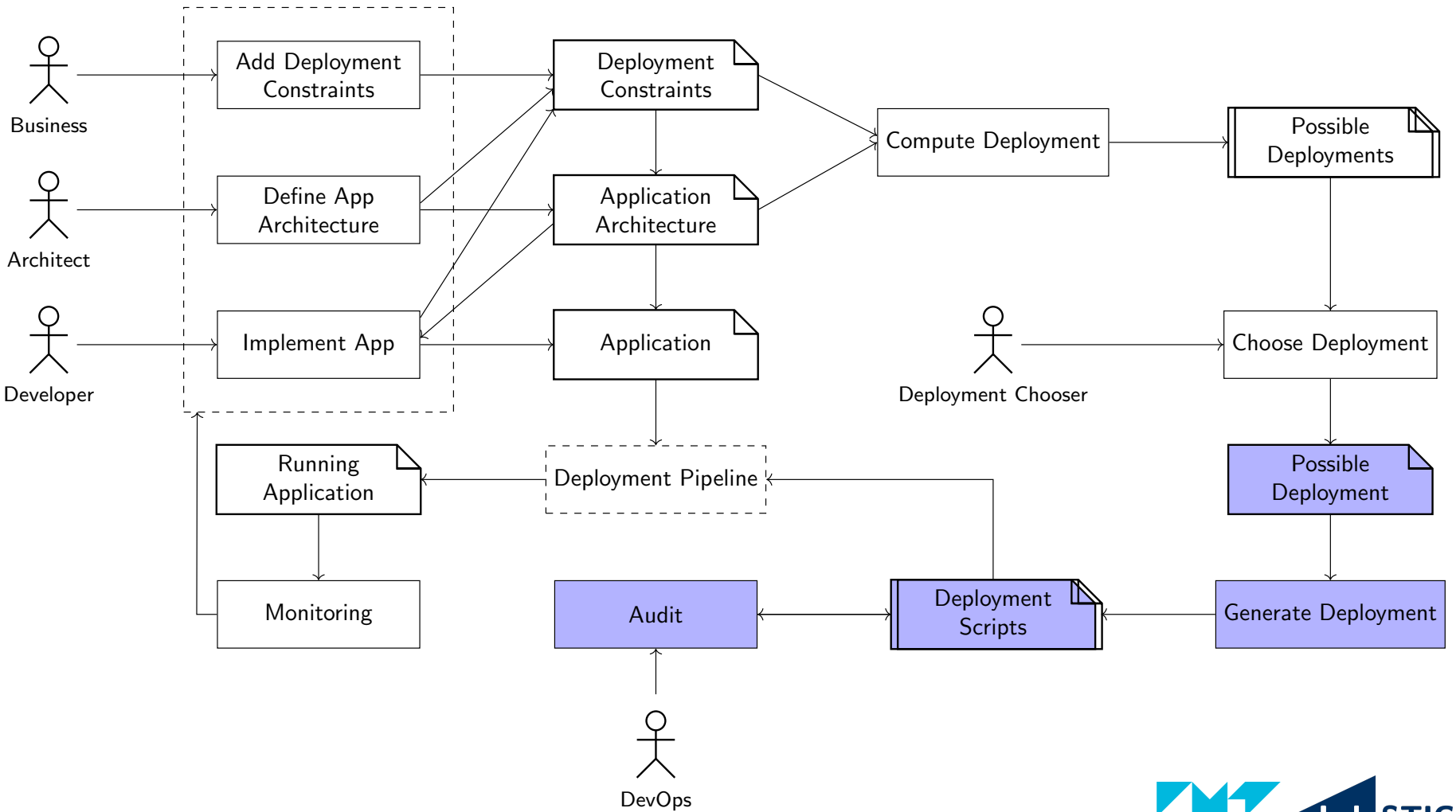
- **Goal:** Compute viable deployment configurations based on defined constraints
- **Who is involved?** Deployment Generator *(partially automated process)*
- **Key Activities:**
  → Analyze infrastructure, security, and business constraints
  → Identify valid deployment configurations
- **Results:**
  → A set of feasible deployment configurations (if any)


Business
Add Deployment Constraints
Deployment Constraints
Architect
Define App Architecture
Application Architecture
Developer
Implement App
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Running Application
Deployment Pipeline
Possible Deployment
Monitoring
Audit
Deployment Scripts
Generate Deployment
DevOps

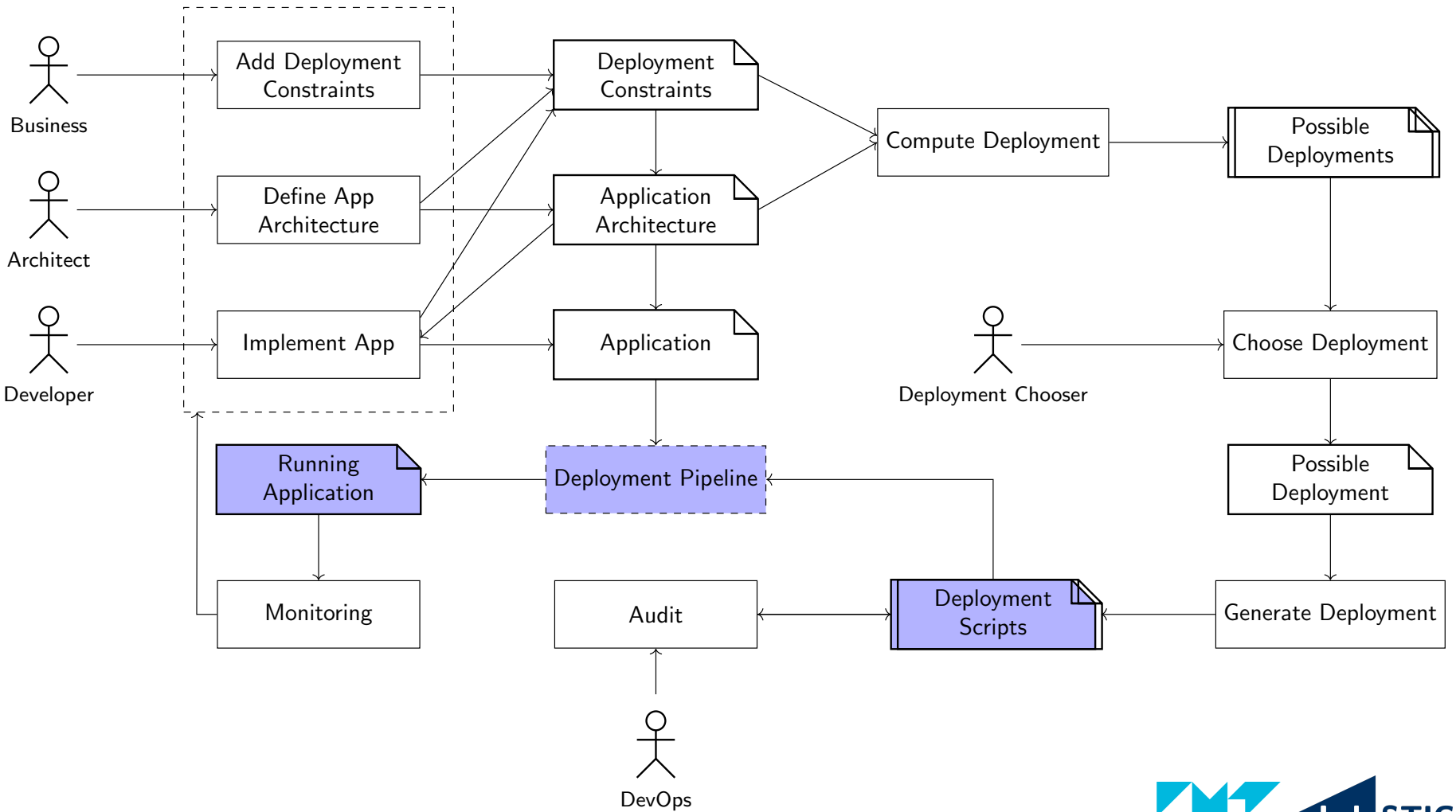
- **Goal:** Choose one deployment from the possible configurations
- **Who is involved?** Deployment Chooser *(partially automated process, human actor)*
- **Key Considerations:**
  → Cost optimization
  → Performance and scalability
  → Security and compliance
- **Results:**
  → A single deployment strategy selected for execution


Business
Architect
Developer
Add Deployment Constraints
Define App Architecture
Implement App
Deployment Constraints
Application Architecture
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Possible Deployment
Running Application
Deployment Pipeline
Monitoring
Audit
Deployment Scripts
Generate Deployment
DevOps

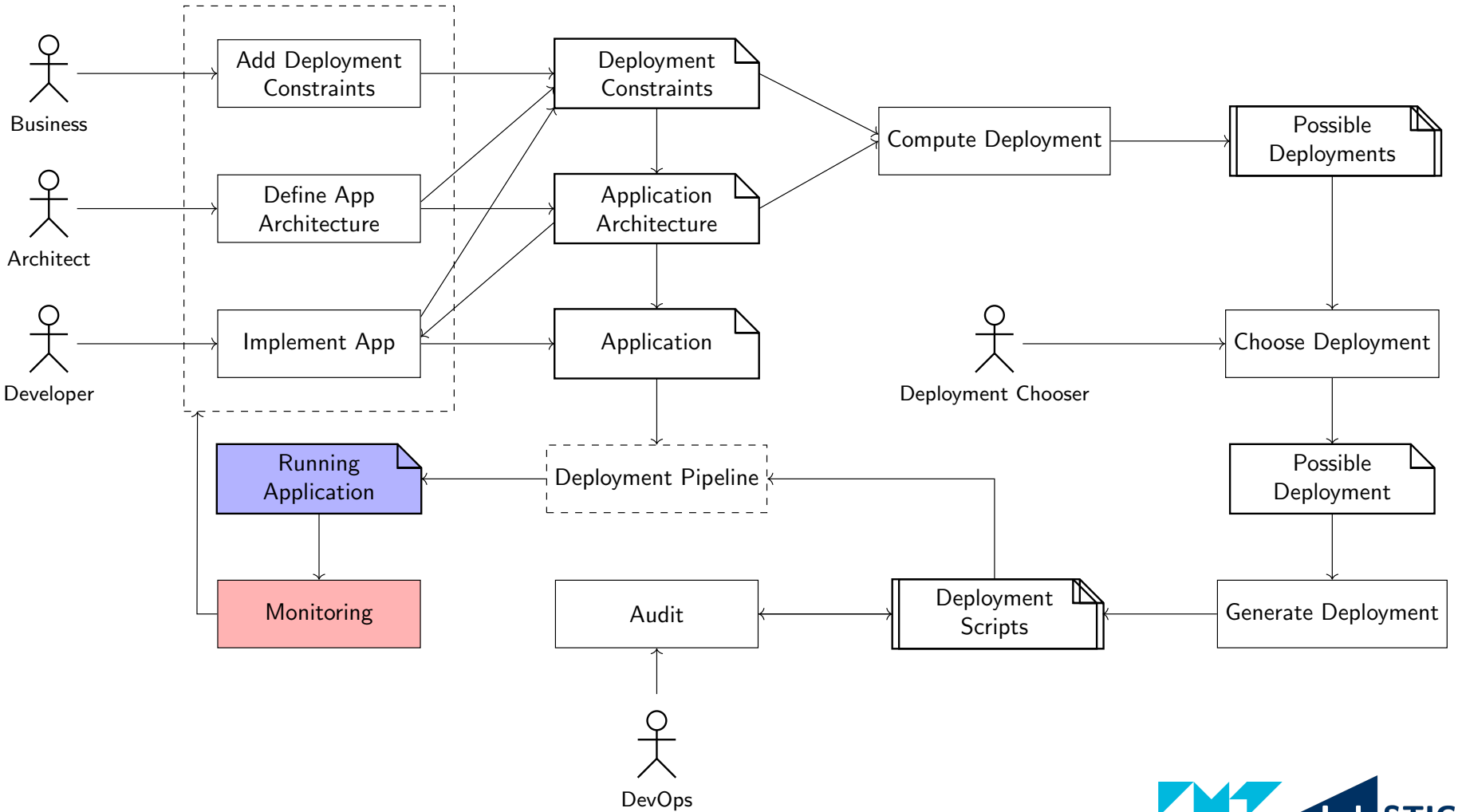
- **Goal:** Generate deployment scripts and verify correctness
- **Who is involved?** Deployment Generator *(automated process)*, DevOps Engineer *(audit)*
- **Key Activities:**
  → Generate deployment scripts (Infrastructure-as-Code)
  → Validate correctness security and compliance of deployment scripts before execution (IaC testing, security scanning)
- **Results:**
  → Ready-to-execute deployment scripts


Business
Architect
Developer
Add Deployment Constraints
Define App Architecture
Implement App
Deployment Constraints
Application Architecture
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Running Application
Deployment Pipeline
Possible Deployment
Monitoring
Audit
Deployment Scripts
Generate Deployment
DevOps

- **Goal:** Execute deployment scripts
- **Who is involved?** Deployment Pipeline *(external process)*
- **Key Activities:**
  → Provision infrastructure
  → Deploy the application
  → Run automated tests to validate deployment
- **Results:**
  → Application is running in the selected environment


Business
Architect
Developer
Add Deployment Constraints
Define App Architecture
Implement App
Deployment Constraints
Application Architecture
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Possible Deployment
Generate Deployment
Running Application
Deployment Pipeline
Monitoring
Audit
Deployment Scripts
DevOps

- **Goal:** Ensure the application runs efficiently and securely
- **Who is involved?** DevOps Engineers, Security Engineers
- **Key Activities:**
  → Performance monitoring (latency, uptime, resource usage)
  → Security monitoring (intrusion detection, compliance audits)
- **Results:**
  → Continuous observability of the deployed application
  → New requirements that will be an input to the process


Business
Architect
Developer
Add Deployment Constraints
Define App Architecture
Implement App
Deployment Constraints
Application Architecture
Application
Compute Deployment
Possible Deployments
Deployment Chooser
Choose Deployment
Possible Deployment
Generate Deployment
Deployment Scripts
Running Application
Deployment Pipeline
Monitoring
Audit
DevOps

A company wants to deploy an e-commerce application that includes a web frontend, a backend service, and a database. The deployment must satisfy the following constraints:

1. Security Constraints:
   - Database: High-security, encryption enabled
   - Backend: Medium-security, only communicates with trusted services
   - Frontend: Lower-security, must enforce HTTPS
2. Resource Constraints:
   - Frontend: $\geq$ 2 vCPUs, 4GB RAM
   - Backend: $\geq$ 4 vCPUs, 8GB RAM
   - Database: $\geq$ 8 vCPUs, 16GB RAM, dedicated storage
3. Cloud Cost Constraints:
   - Choose the cheapest provider meeting all constraints
   - Prefer reserved instances over shared ones
   - Deploy in cost-effective regions

1.A. Define Application Architecture & Constraints:

- Identify frontend, backend, and database as separate components
- Specify security, cost, and resource constraints

1.B. Implement the Application:

- Construct deployable artifacts

2. Compute Possible Deployments:

- Evaluate all of the specified constraints
- Identify the cheapest region meeting all constraints

3.A. Choose a Deployment:

- Balance security, cost, and resource configuration

3.B. Generate & Audit Deployment Scripts:

- Generate IaC scripts for selected providers
- Audits ensure that the generated IaC scripts align with the requirements and are free from security-related issues

4.A. Deploy the Application & Infrastructure:

- Execute deployment scripts through CI/CD pipeline

4.B. Monitor Operations & Security:

- Performance (CPU/RAM usage)
- Security (Unauthorized access)
- Cost (Budget compliance)

Feature	**Traditional Deployment**	**Constraint-Driven Deployment**
Deployment Process	Manual & Script-Based	Computed
Security Consideration	Added Later	Integrated Early
Flexibility	Limited	Adaptive
Error-Prone	Yes	Reduced Errors
Compliance to Constraints	Must be Checked	By-Design

- Benefits:
  - → Integrates with existing deployment pipelines *(CI/CD pipelines)*
  - → Allows different kinds of actors to collaborate in the process of deployment

- Limitations:
  - → Expands the responsibilities of architects and other stakeholders
  - → Introduces new tools and languages, potentially increasing complexity

→ Why define a deployment DSL?
- Deployment constraints must be precisely specified.
- Automating deployments requires a structured language.

→ **Solution**: A domain-specific language to express constraints

→ Main meta-model entities:
- → `AbstractApplication`, `AbstractComponent`, `AbstractService`, `Context`
- → `ApplicationImplementation`, `ComponentImplementation`, `ServiceImplementation`, `Resource`
- → `ApplicationConstraint`, `ComponentConstraint`, `ServiceConstraint`, `ContextConstraint`
- → `RunningApplication`, `RunningService`, `RunningComponent`, `RunningResource`,

- Refine the meta-model to accurately represent real-world deployment scenarios
- Develop the deployment generator:
  - → Define basic constraints
  - → Compute possible deployment configurations based on the specified constraints
- Generate infrastructure-as-code (IaC) deployment scripts for the computed configurations