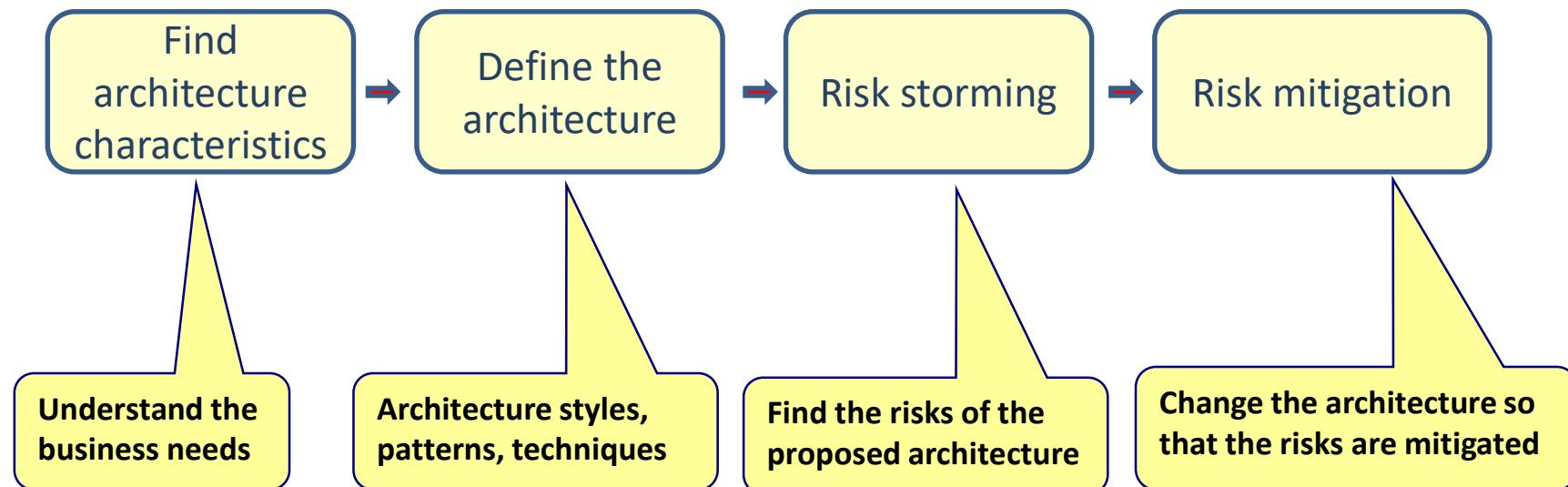


Lesson 14

# **ARCHITECTURE ANALYSIS**

# Software architecture

---



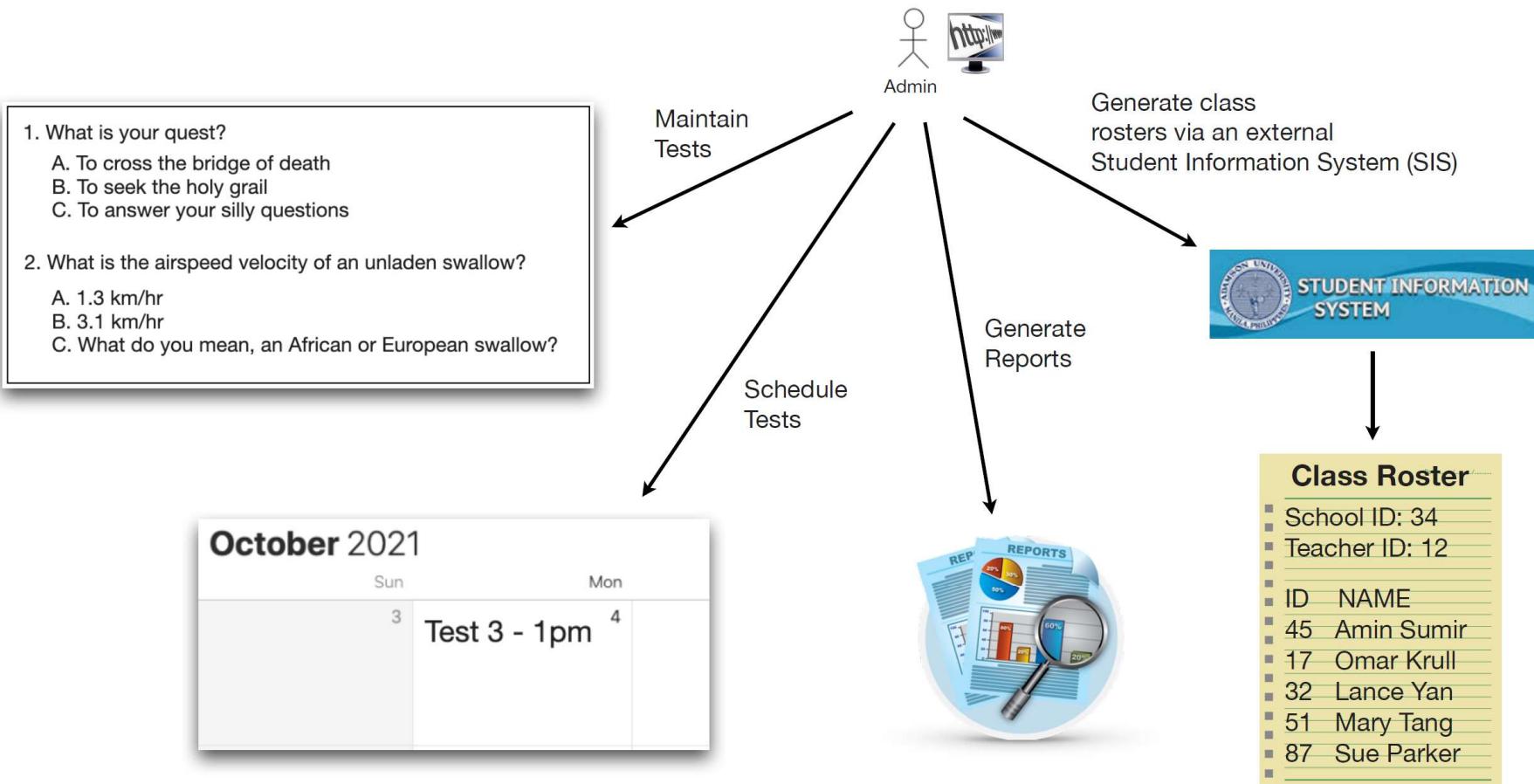
# Standardized testing system

---

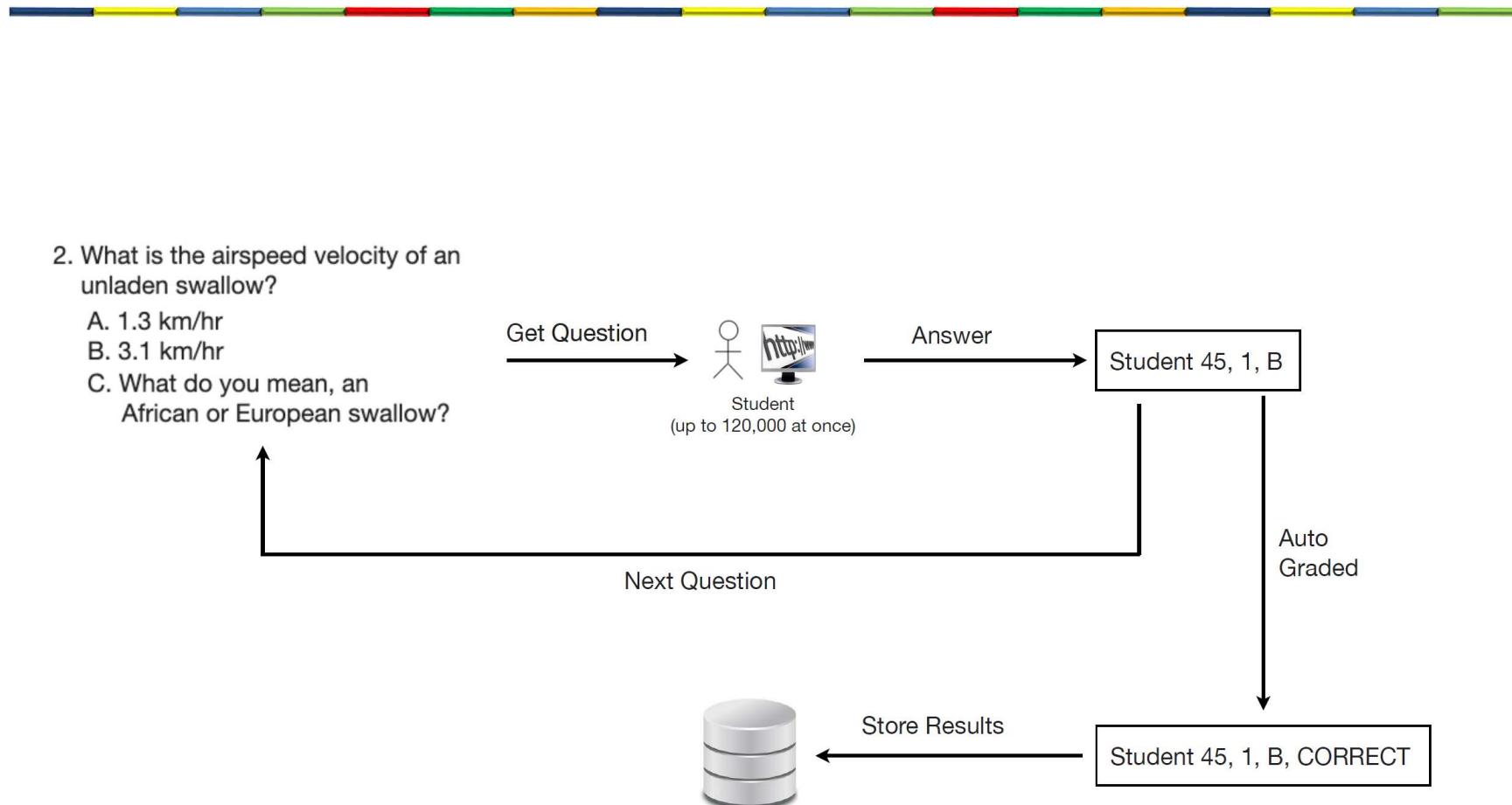
Your state (or country) would like a new system to support standardized testing for all public and private schools, grades 3-12.

- Requirements:
  - Students will use a new web-based (and backend) testing application to take the test from their school based on their assigned teacher. Questions are presented to the student, they submit the answer (true/false or multiple choice), and get the next question. The student answers the question and the corresponding grade for each question (right or wrong) are eventually consolidated into a single test answer database representing all of the test scores for all students.
  - Administrative staff will use the system to create the test and corresponding answer keys, schedule the tests, and generate the sign-in roster from an existing external Student Information System (SIS). After the testing is complete, the admin staff will use the new system to generate student, teacher, school, and question validation reports.
  - Users: 800,000+ students (up to 120,000 at a time), 2 administrators (test creation, test scheduling, etc.)
- Additional info from the education department:
  - “It is ***absolutely imperative*** that no student answers are lost in the event of a system crash.”
  - “We need the new system before the start of the next term (6 months).”
  - “It is vital that the test answers be protected from prying eyes.” “We need to make sure a student doesn’t take the test multiple times.”
  - “Somehow we need to make sure a student doesn’t take a test for someone else.”

# Admin user



# Student user



# Architectural characteristics

Your state (or country) would like a new system to support standardized testing for all public and private schools, grades 3-12.

- Requirements:

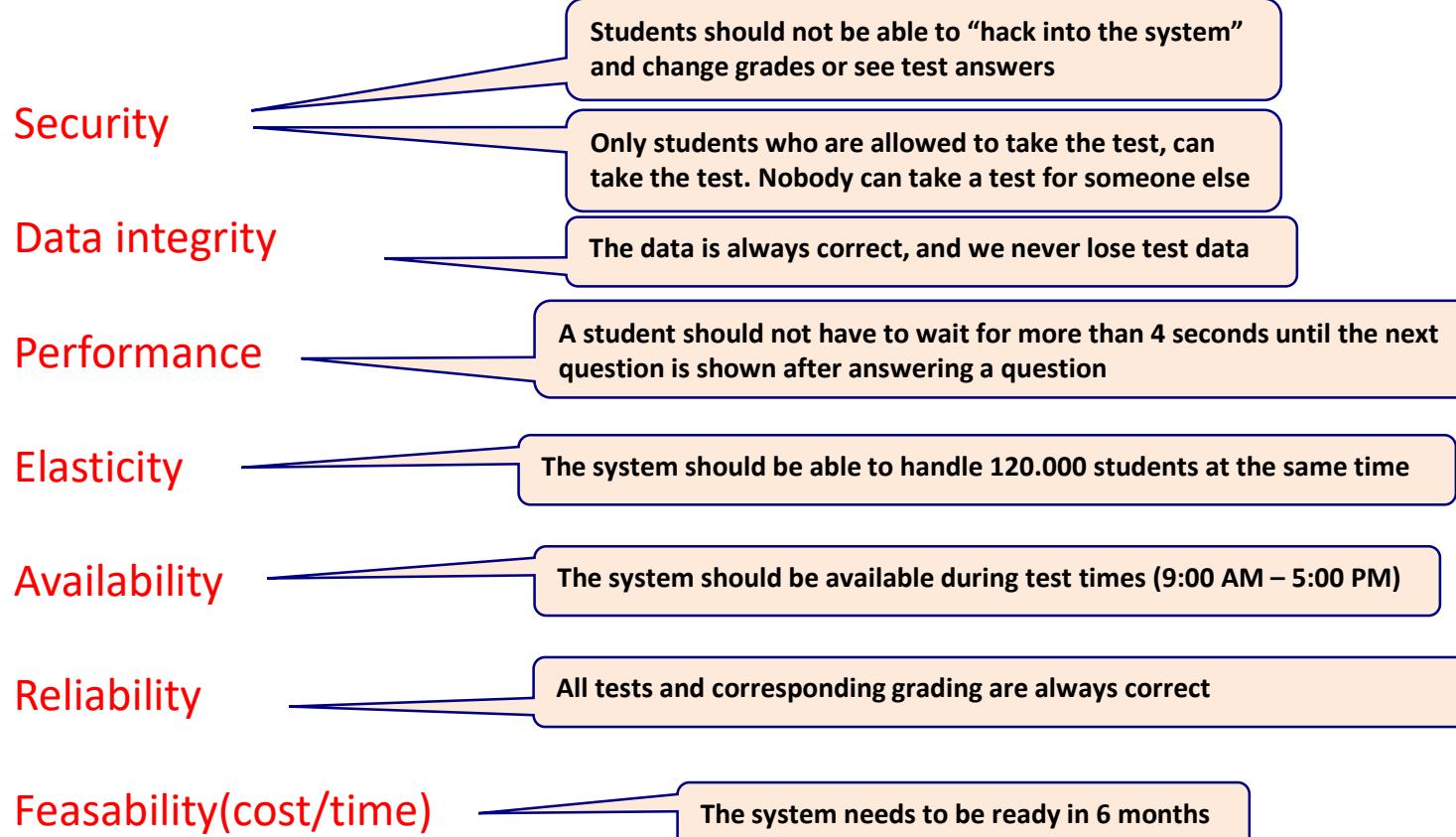
- Students will use a new web-based (and backend) testing application to take the test from their school based on their assigned teacher. Questions are presented to the student, they submit the answer (true/false or multiple choice), and get the next question. The student answers the question and the corresponding grade for each question (right or wrong) are eventually consolidated into a single test answer database representing all of the test scores for all students.
- Administrative staff will use the system to create the test and corresponding answer keys, schedule the tests, and generate the sign-in roster from an existing external Student Information System (SIS). After the testing is complete, the admin staff will use the new system to generate student, teacher, school, and question validation reports.
- Users: 800,000+ students (up to 120,000 at a time) 2 administrators (test creation, test scheduling, etc.)

- Additional info from the education department:

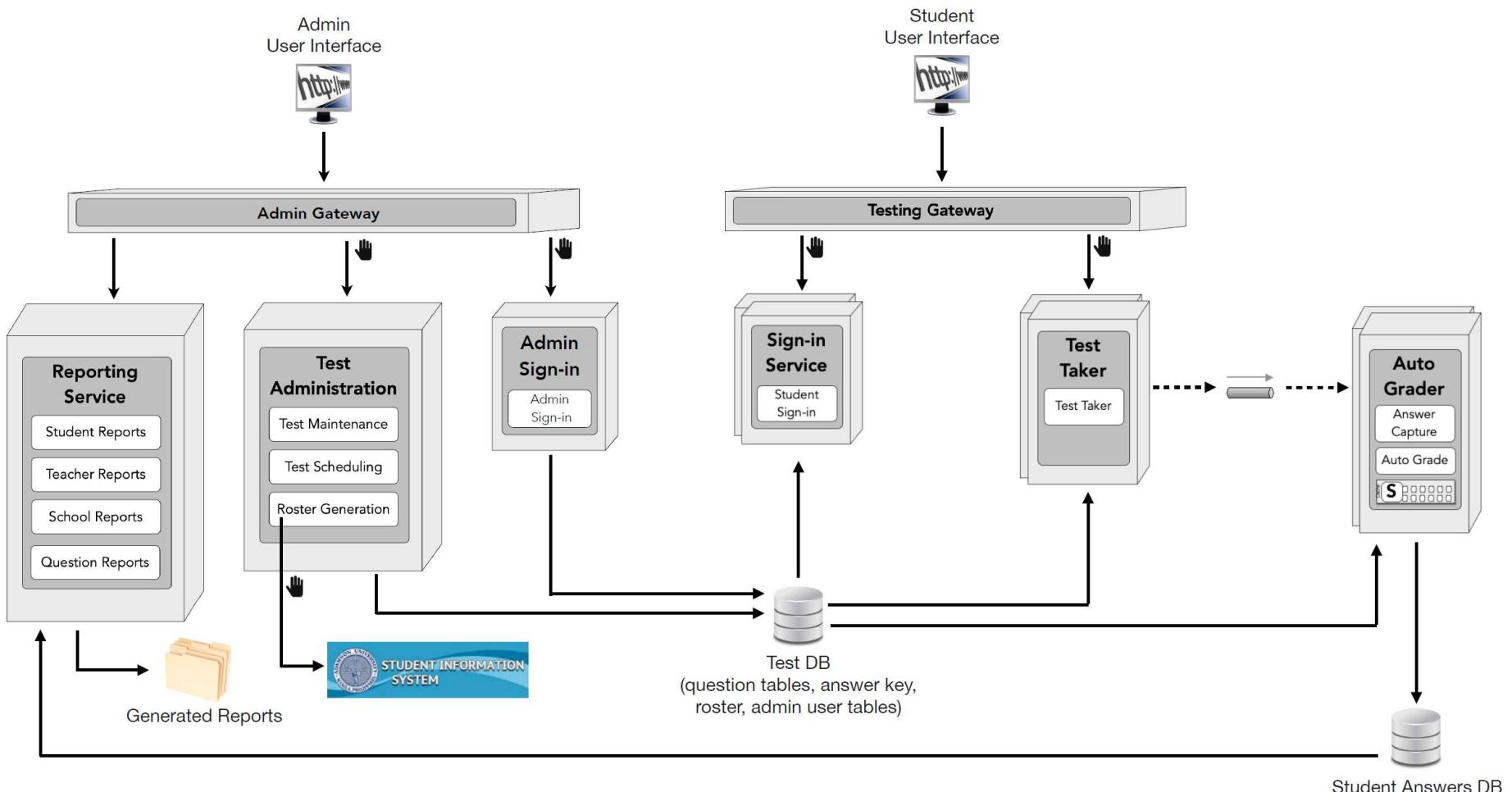
- “It is ***absolutely imperative*** that no student answers are lost in the event of a system crash.”
- “We need the new system before the start of the next term (6 months).”
- “It is vital that the test answers be protected from prying eyes.” “We need to make sure a student doesn’t take the test multiple times.”
- “Somehow we need to make sure a student doesn’t take a test for someone else.”

# Architectural characteristics

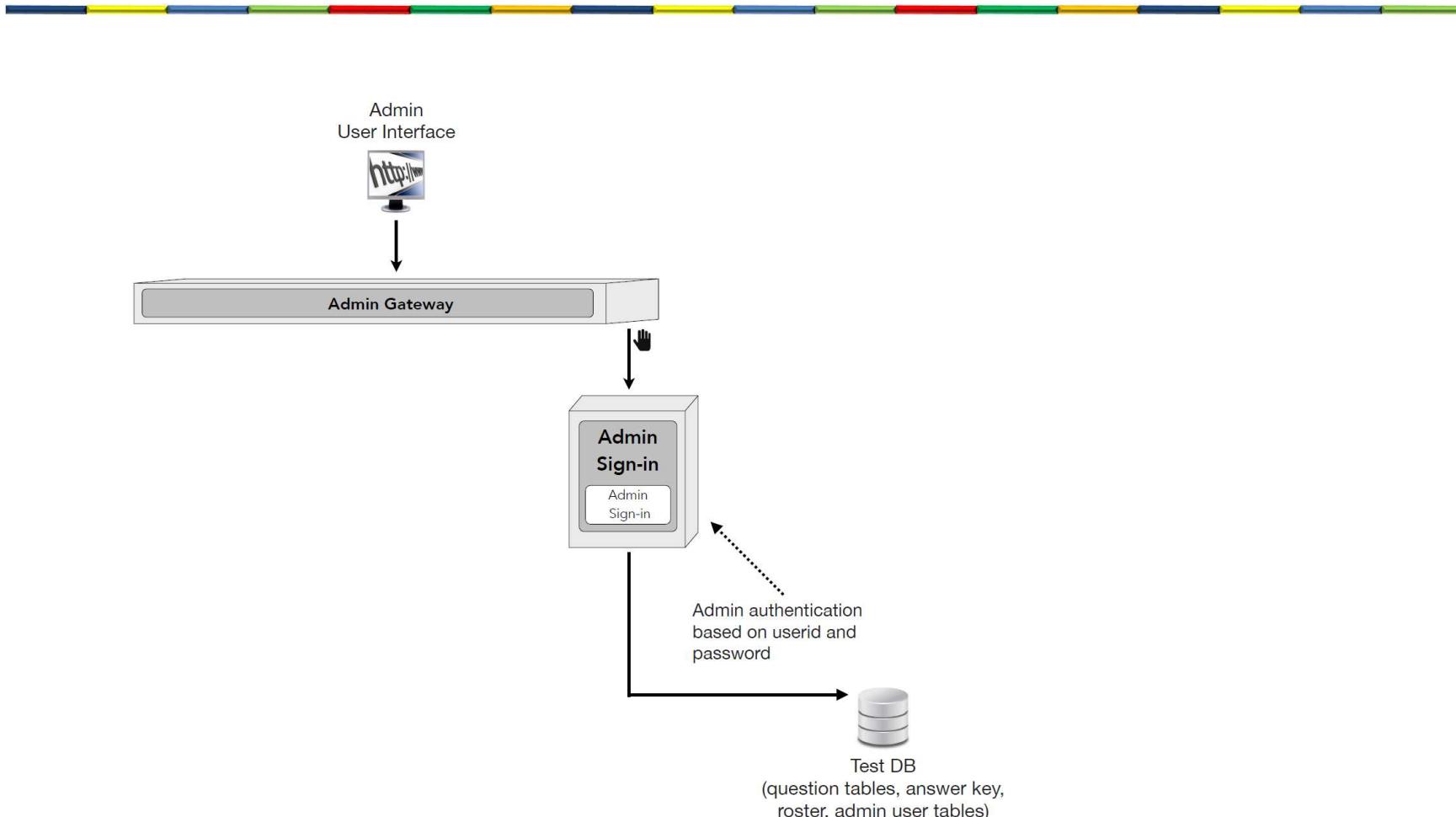
---



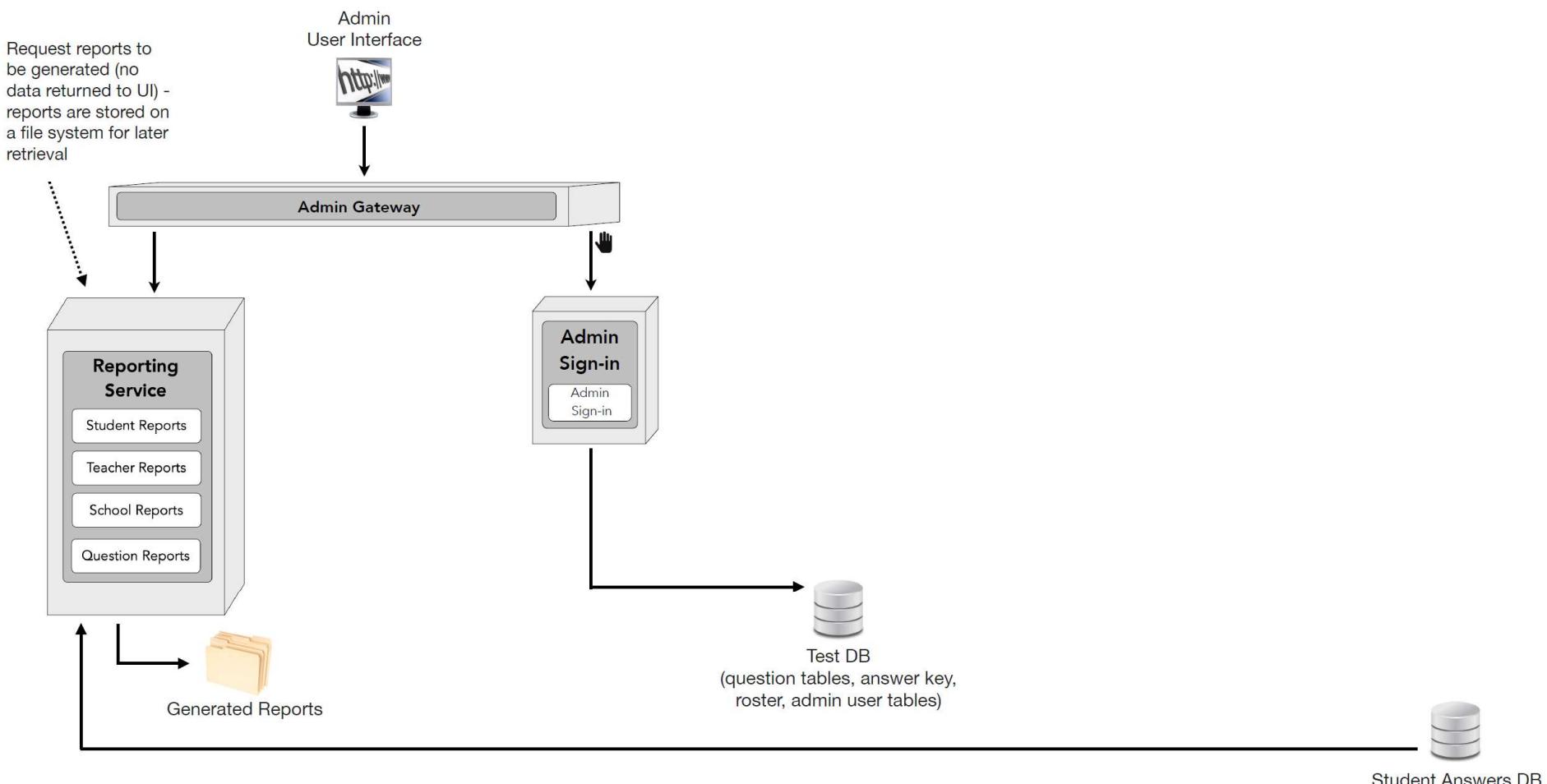
# Proposed architecture



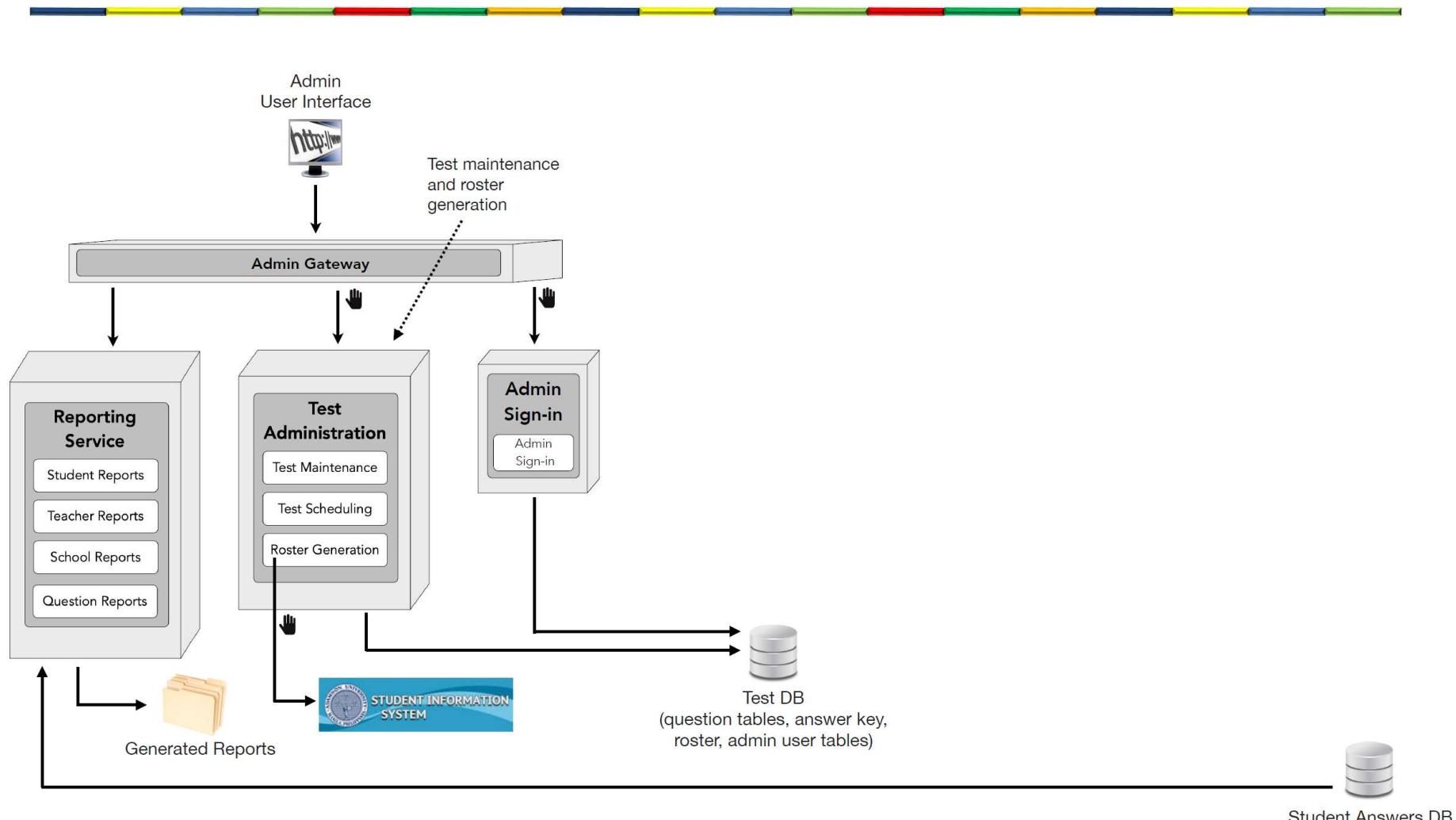
# Admin sign-in



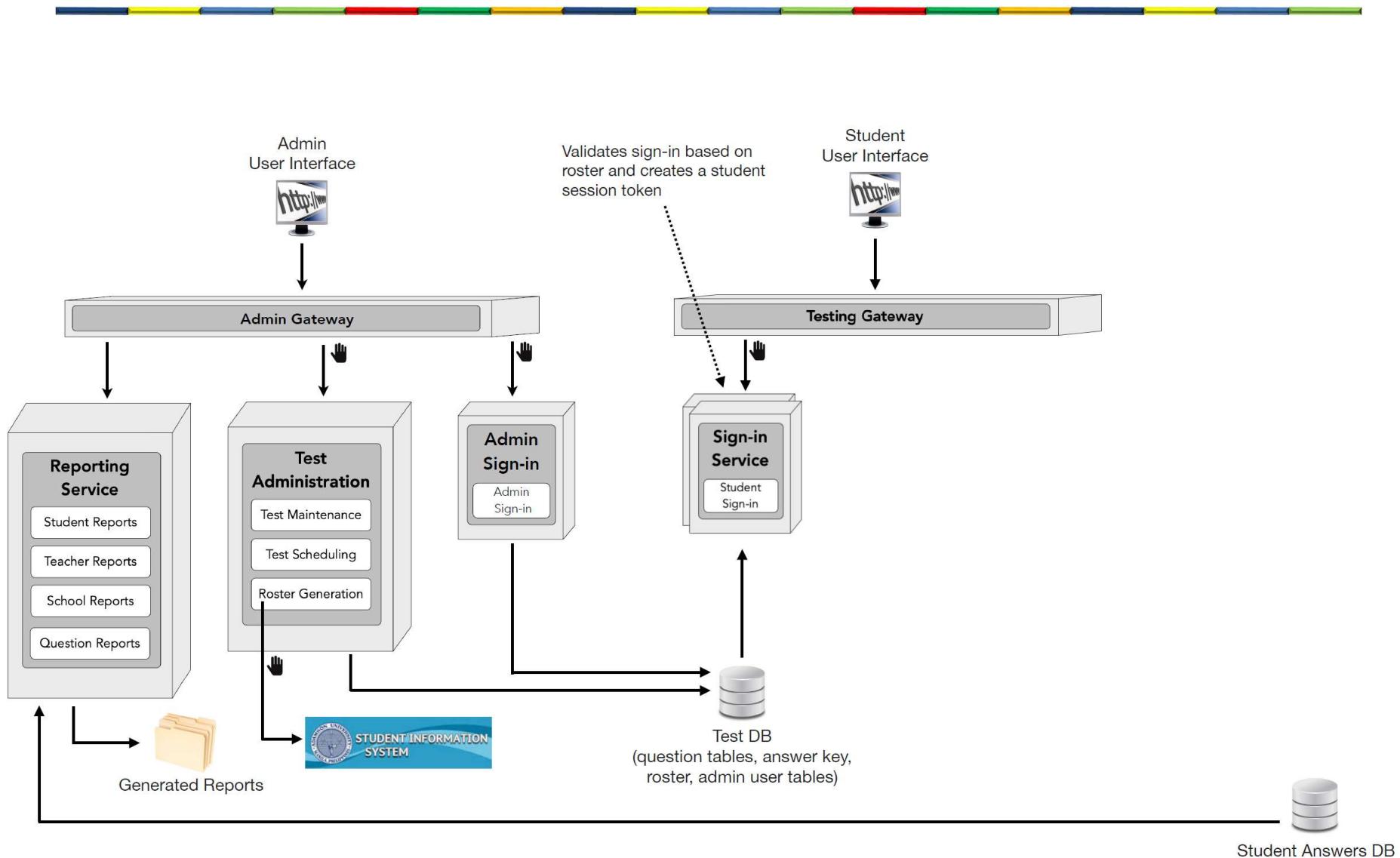
# Reporting



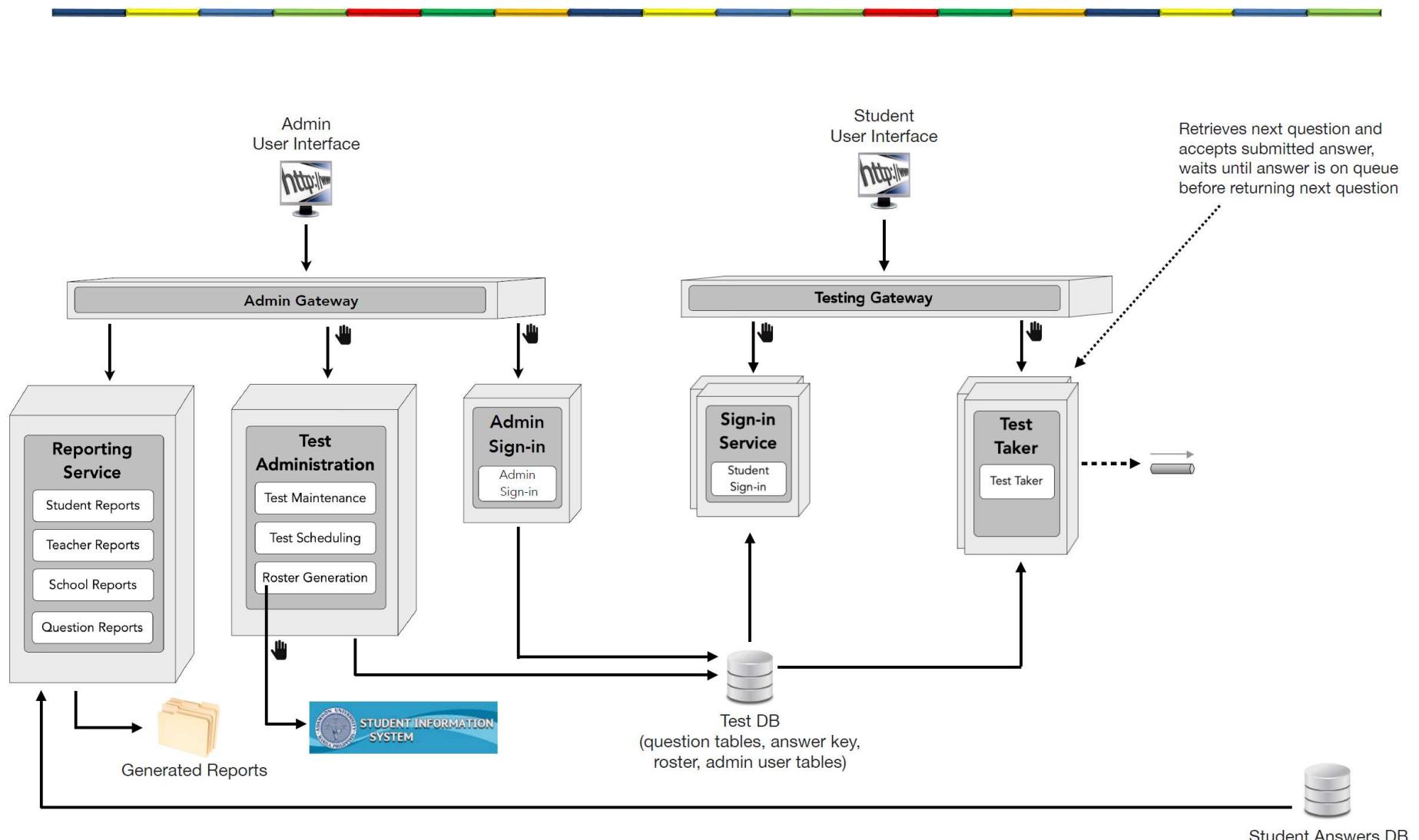
# Test administration



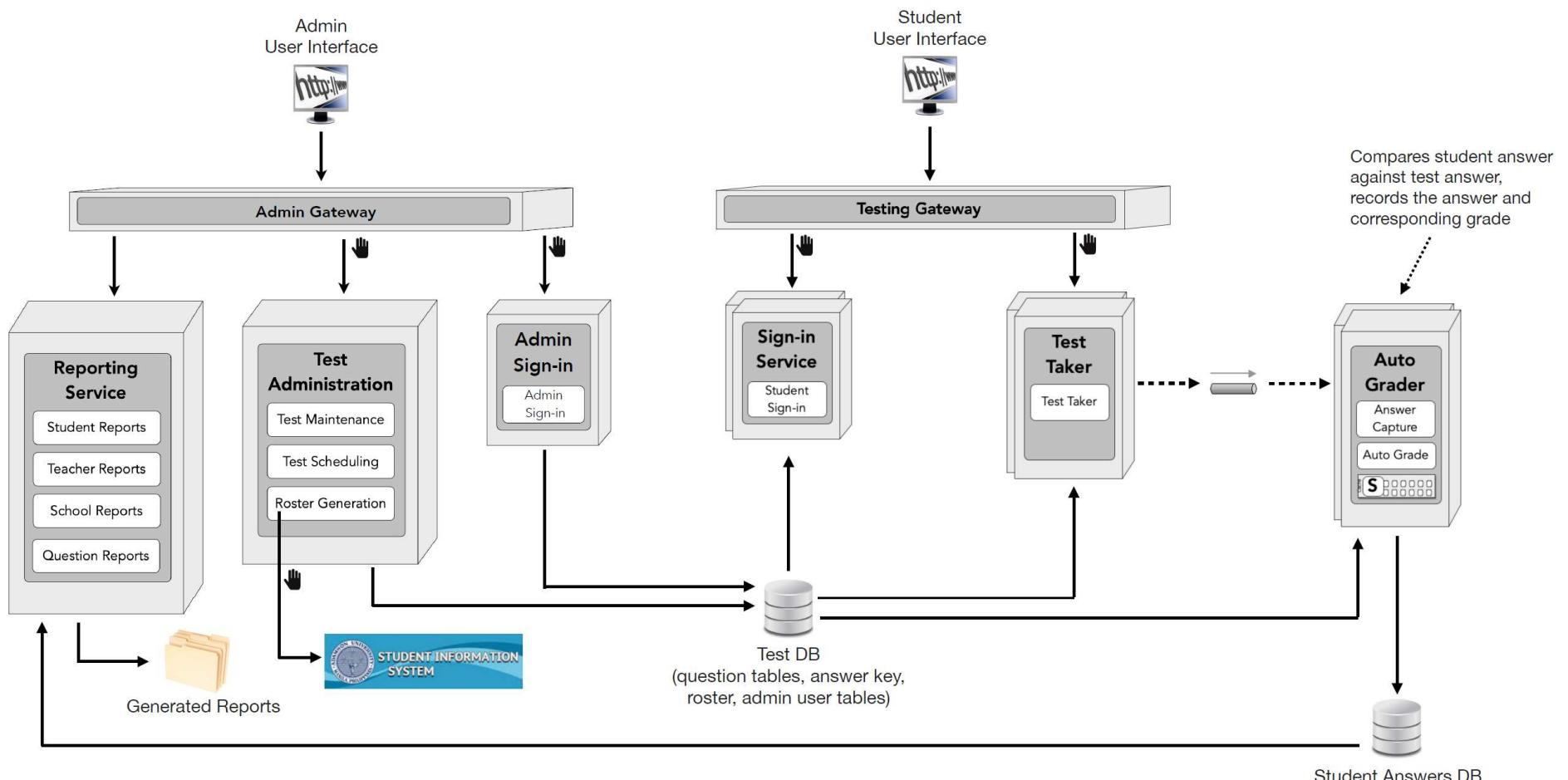
# Student sign-in



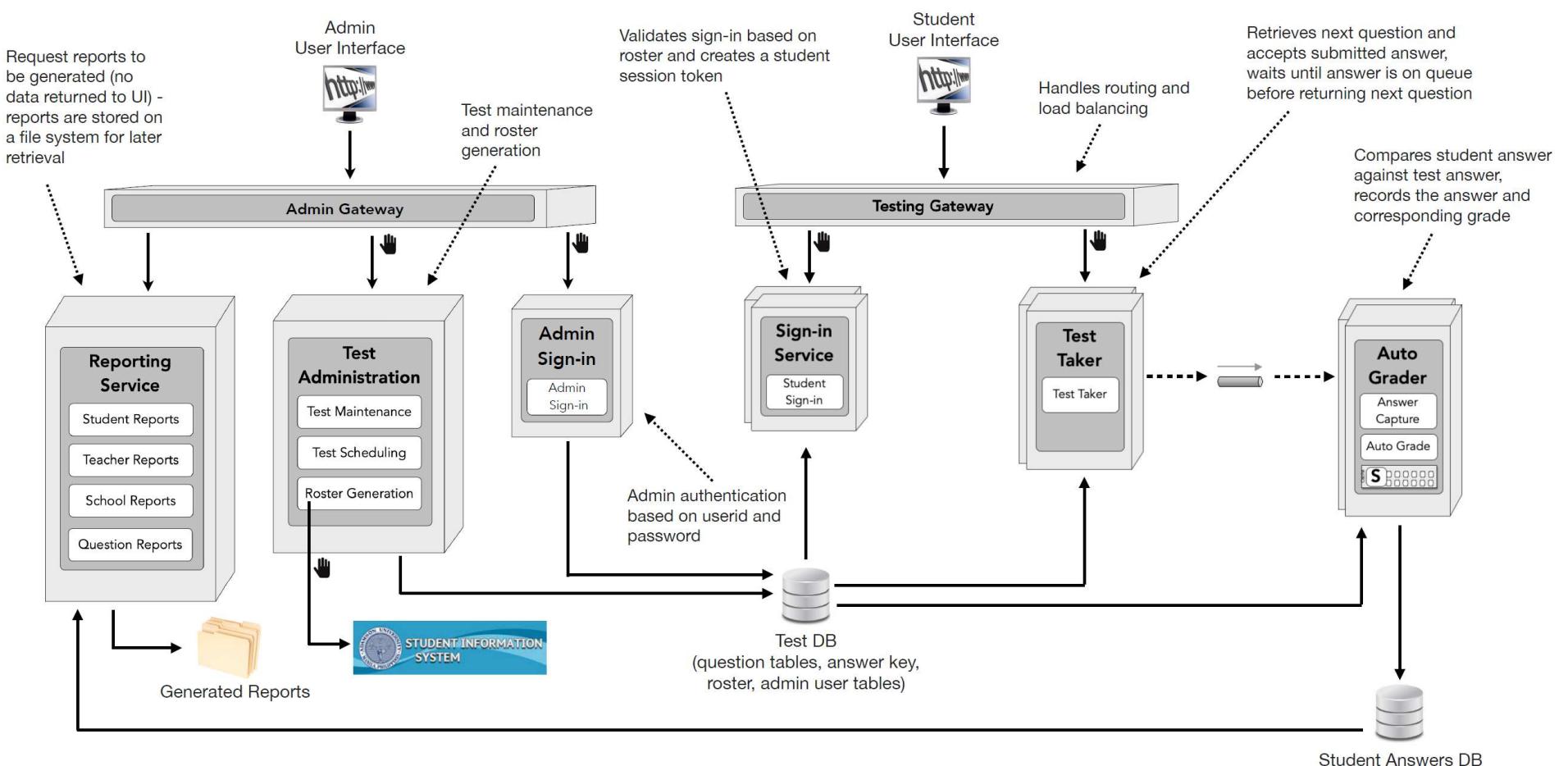
# Students take the test



# Automated grading



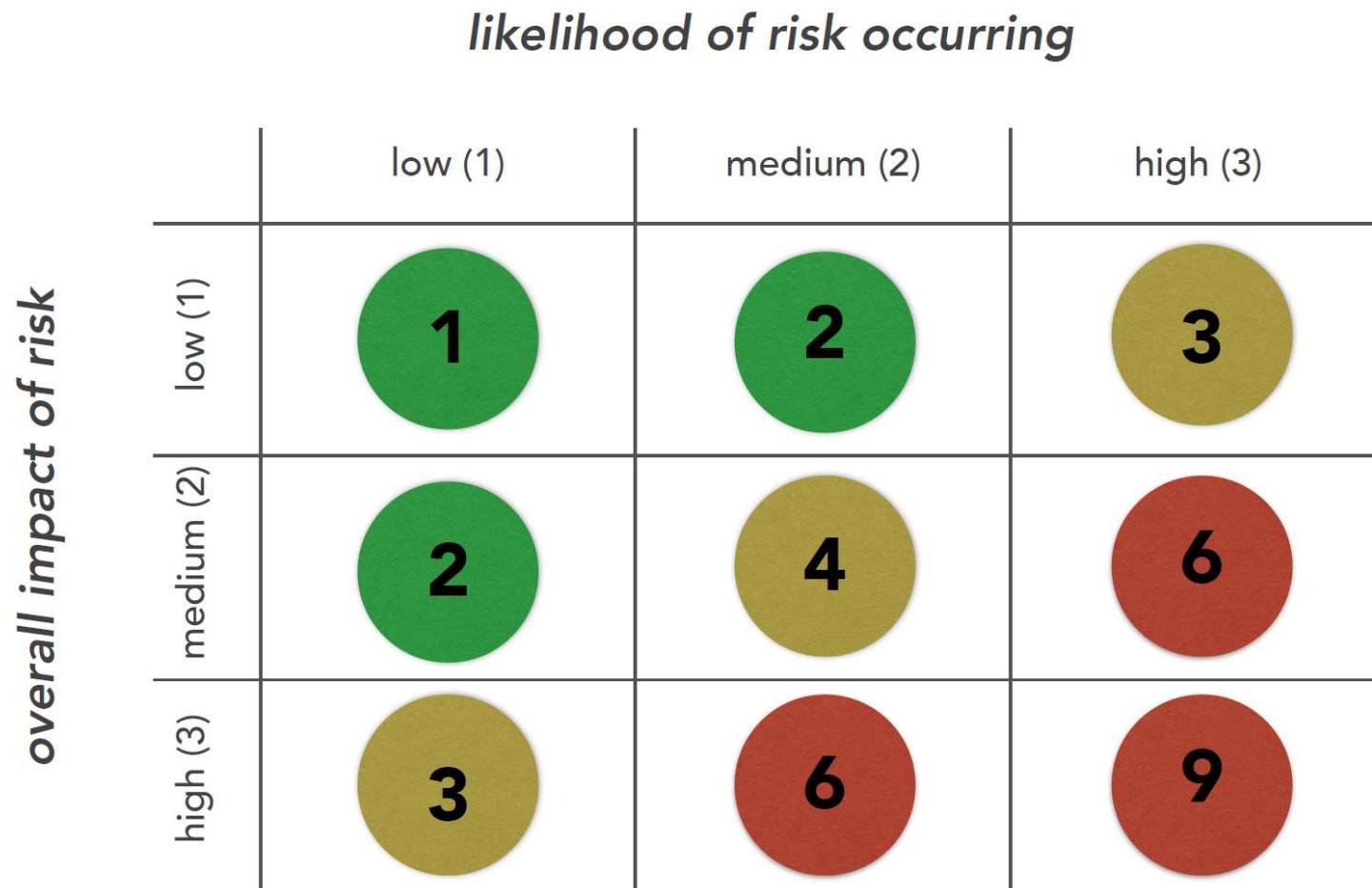
# Proposed architecture



# **RISK STORMING**

# Quantify risk

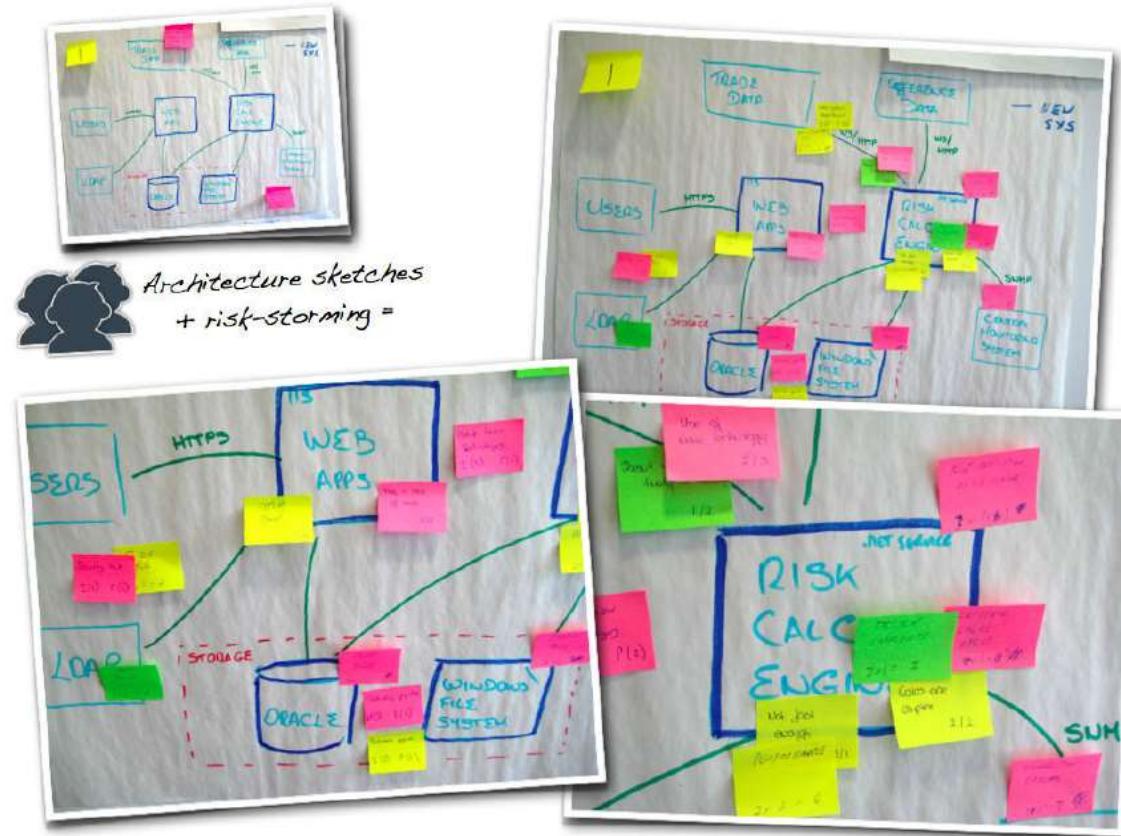
---



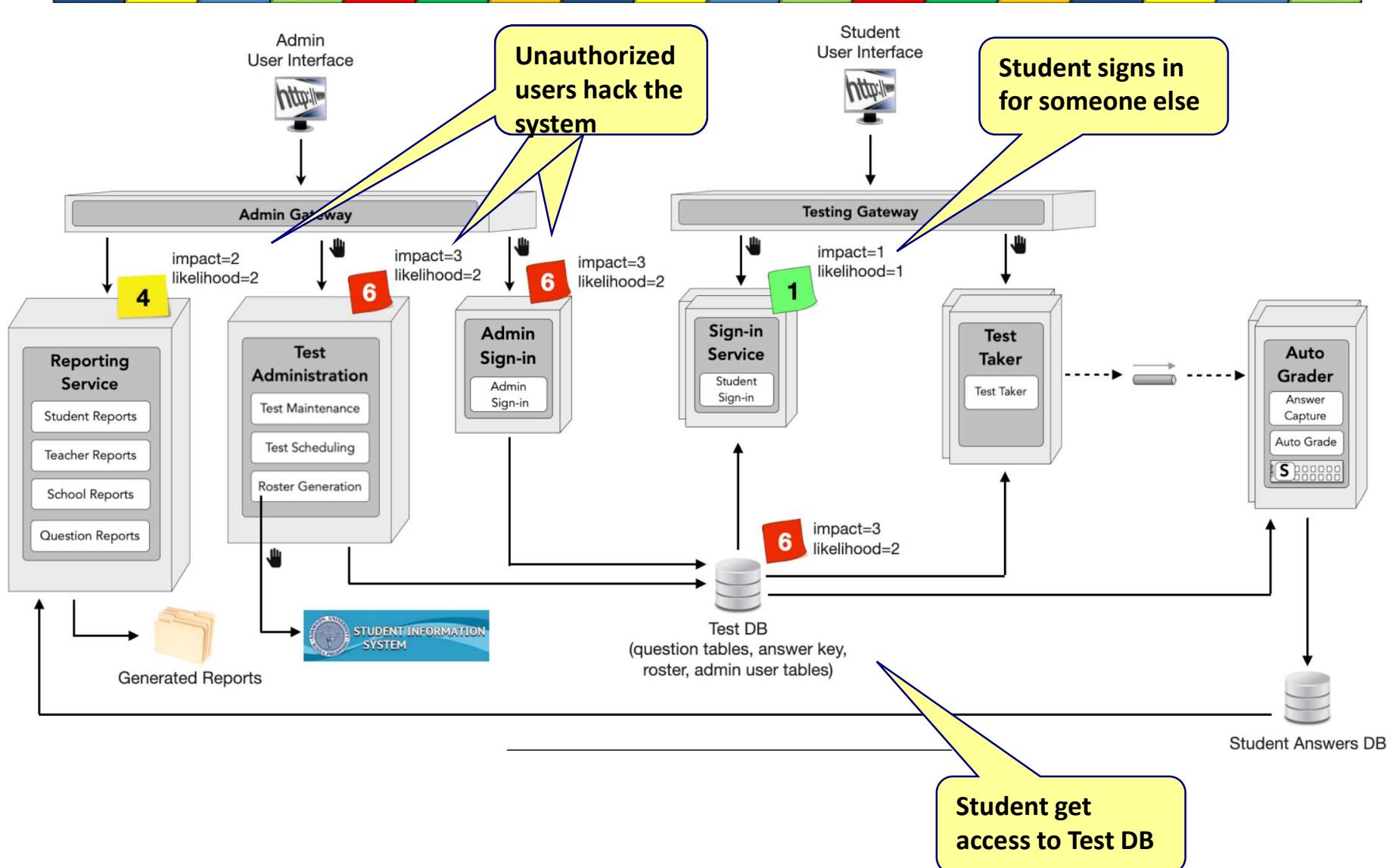
# Risk storming

---

- Quick and fun technique that provides a collaborative way to identify and visualize risks.



# Security risk

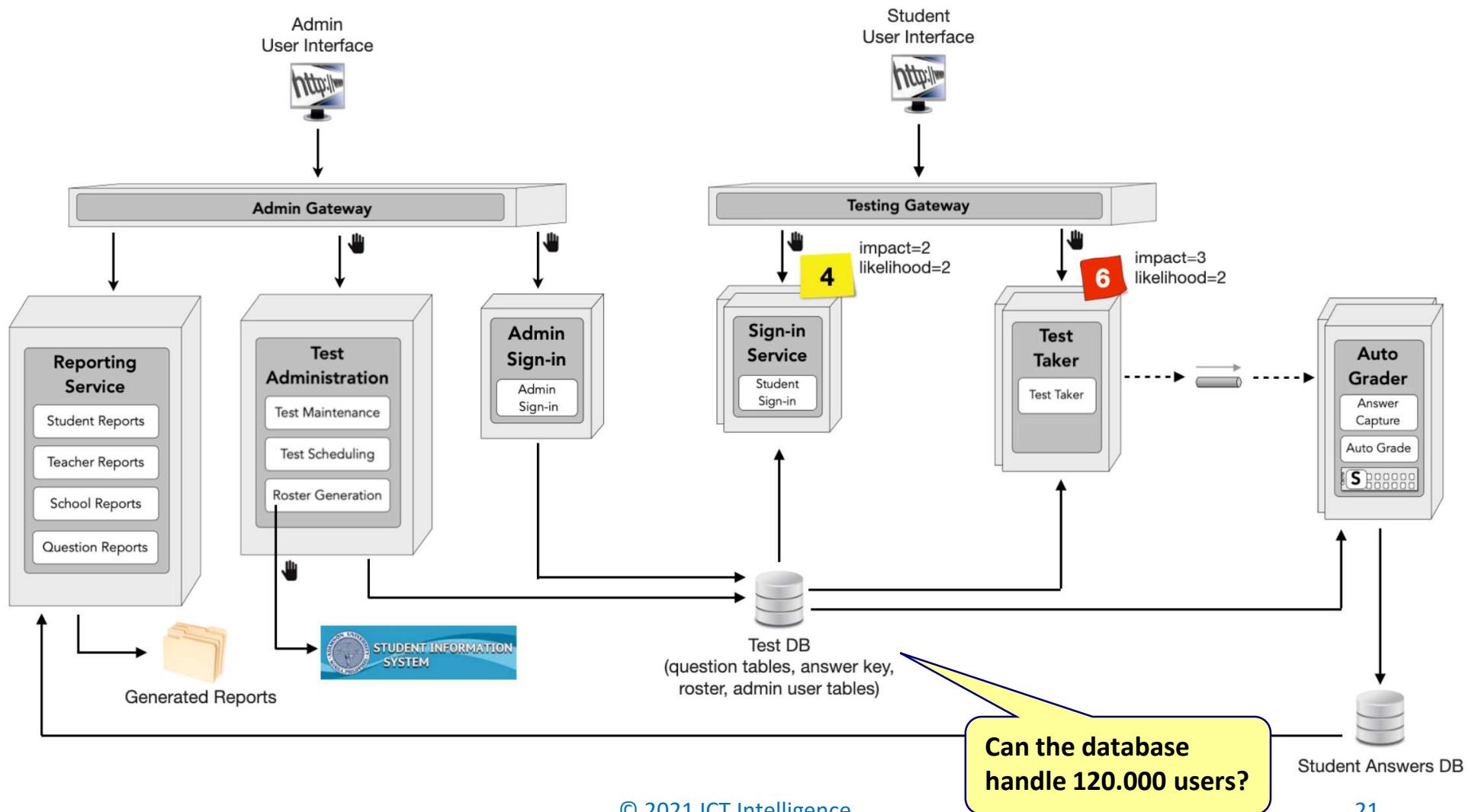


# Security risk



Architecture Risk Assessment	Test Administration	Test Taking	Grading
Data Integrity			
Responsiveness			
Elasticity			
Security	6	6	1
Availability			
Feasibility (cost/time)			
Reliability			

# Responsiveness risk

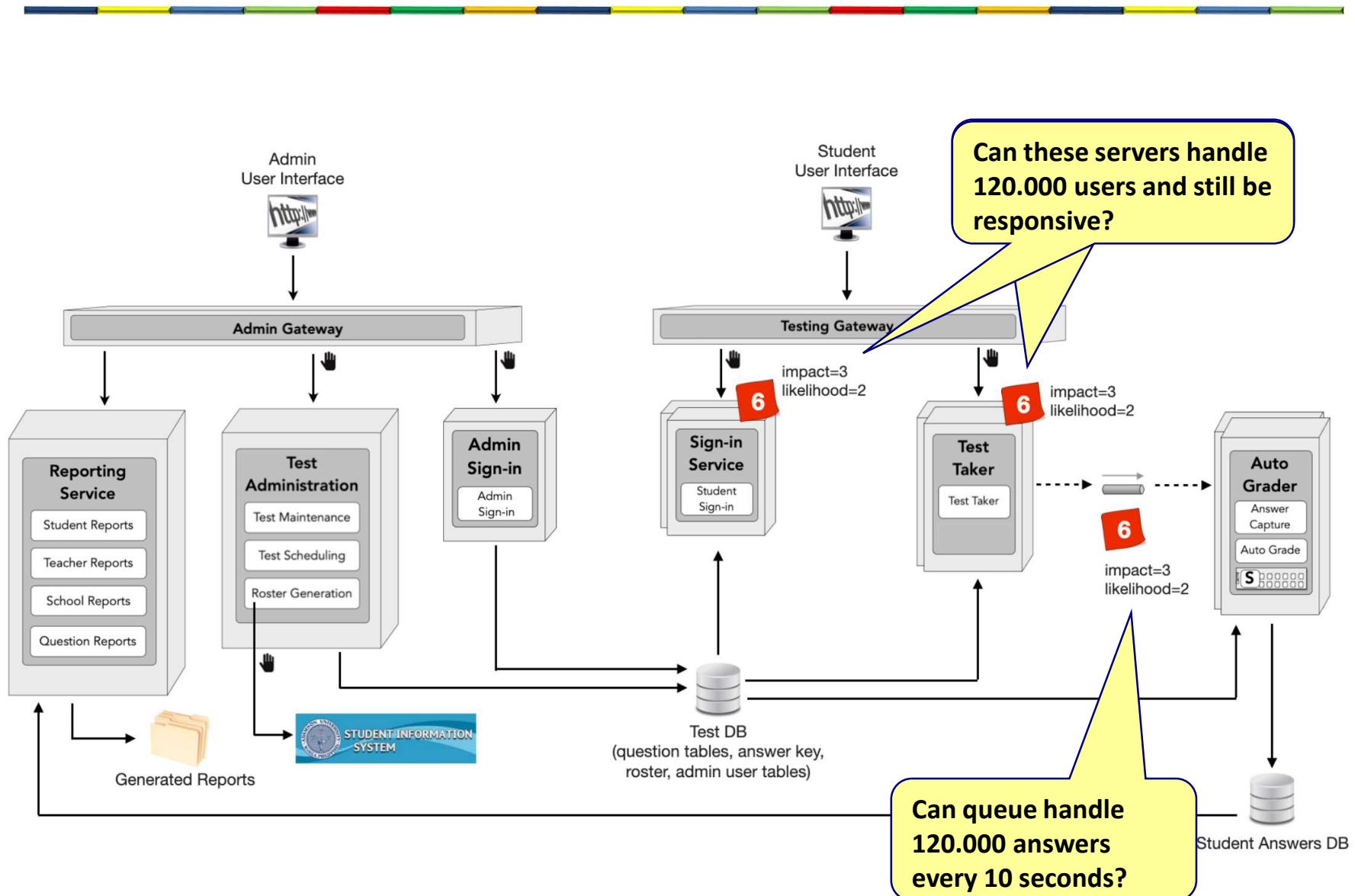


# Responsiveness risk



Architecture Risk Assessment	Test Administration	Test Taking	Grading
Data Integrity			
Responsiveness	1	6	1
Elasticity			
Security	6	6	1
Availability			
Feasibility (cost/time)			
Reliability			

# Elasticity risk

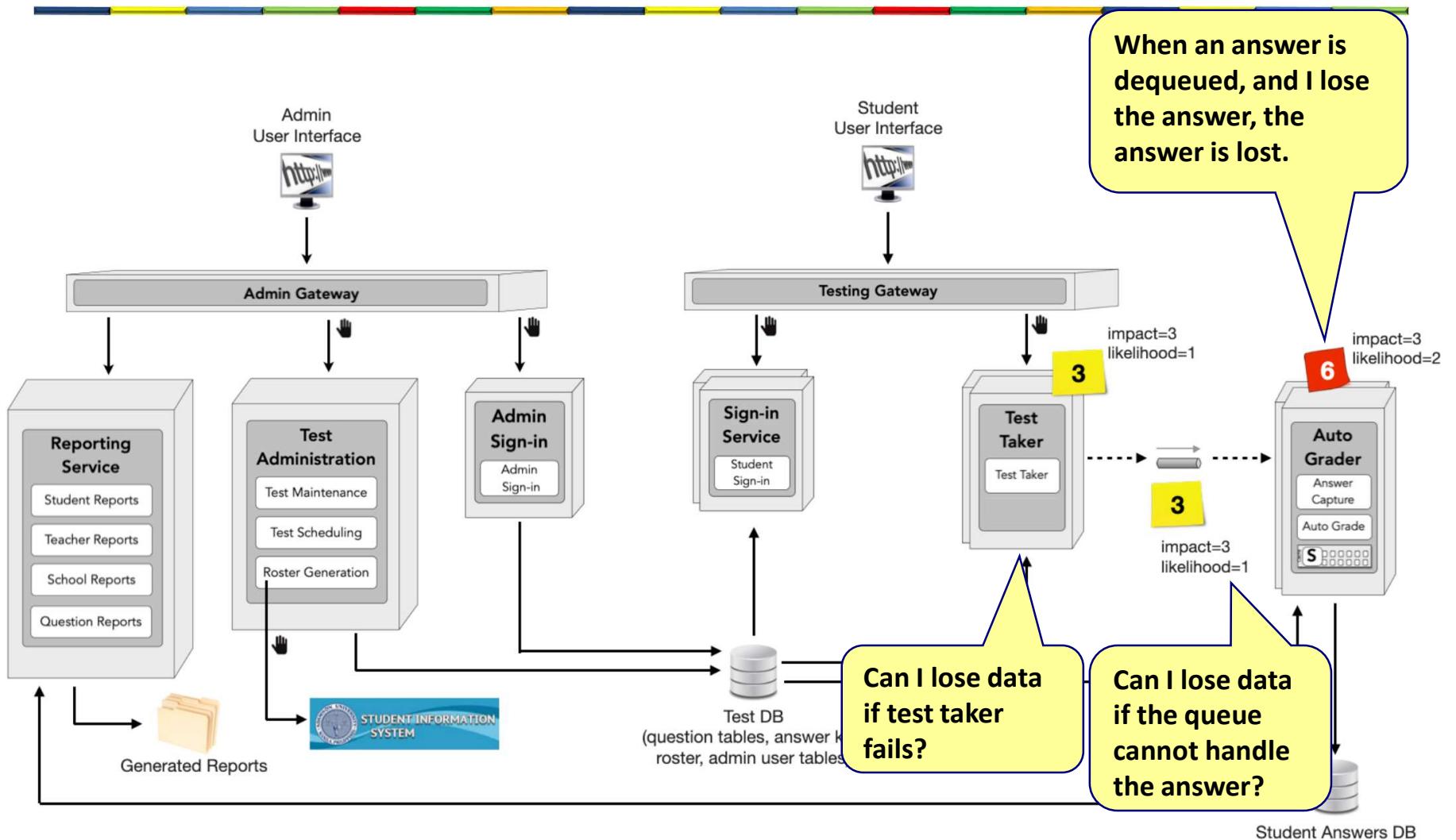


# Elasticity risk



Architecture Risk Assessment	Test Administration	Test Taking	Grading
Data Integrity			
Responsiveness	1	6	1
Elasticity	1	6	1
Security	6	6	1
Availability			
Feasibility (cost/time)			
Reliability			

# Data integrity risk



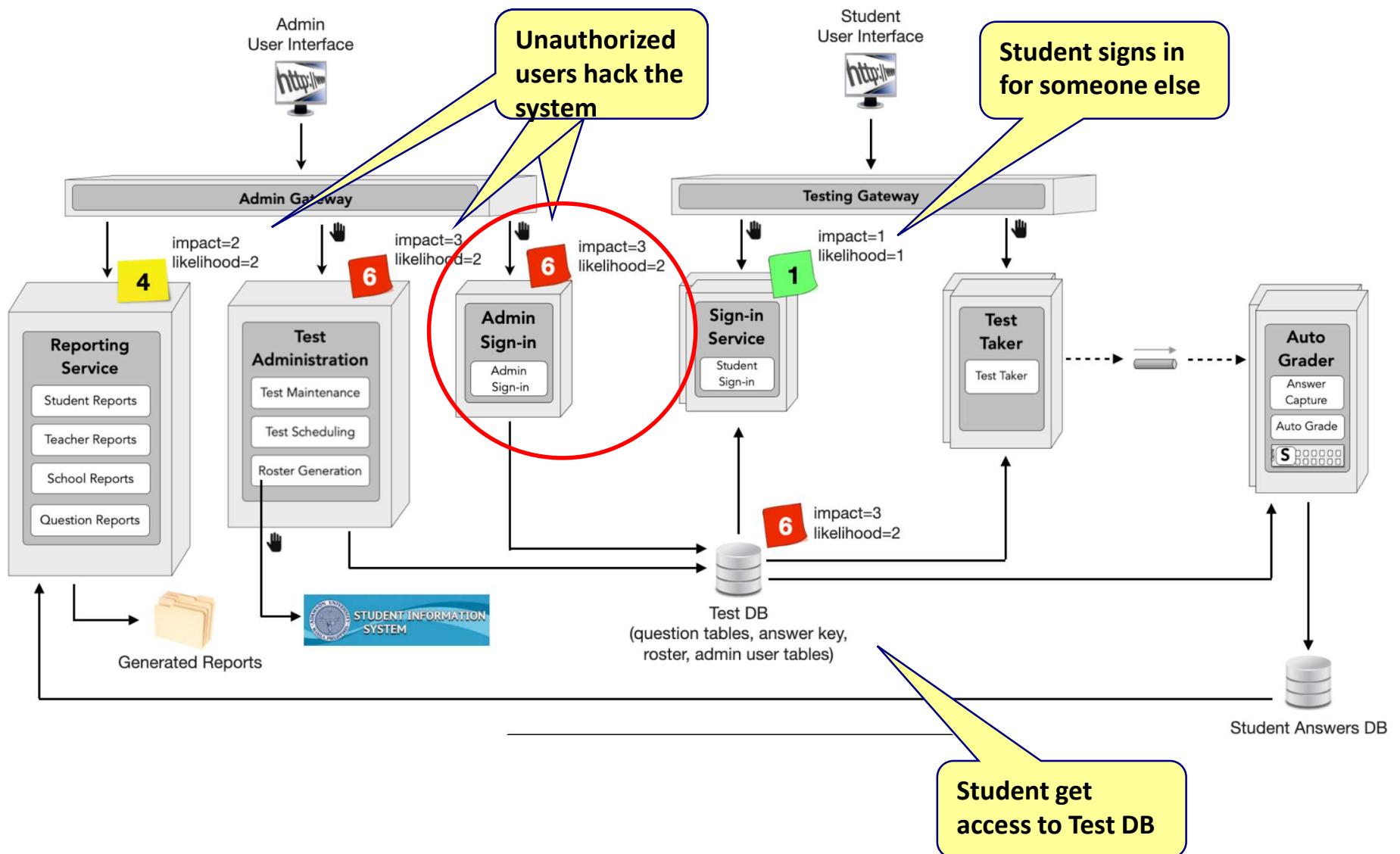
# Risk storming result



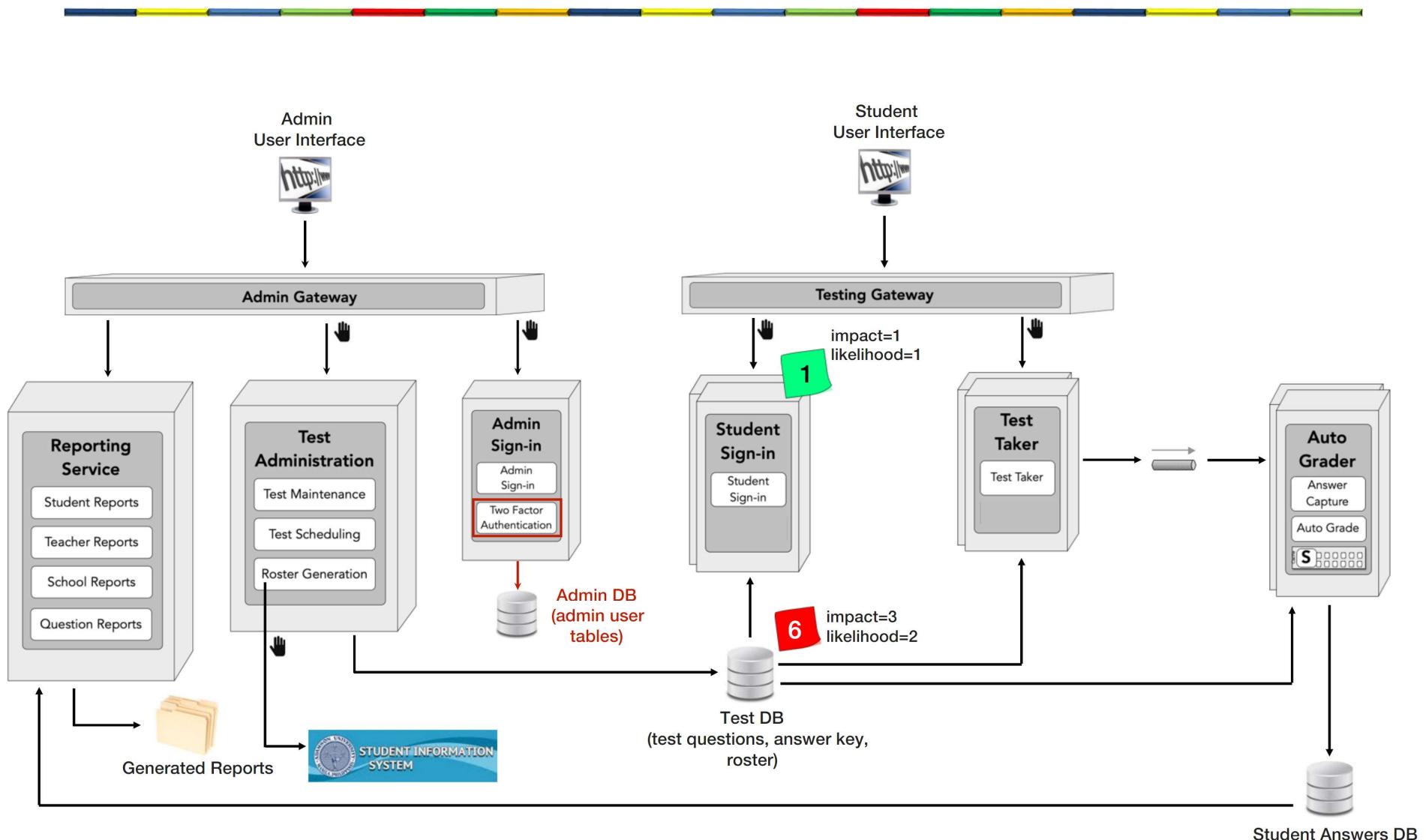
Architecture Risk Assessment	Test Administration	Test Taking	Grading
Data Integrity	1	3	6
Responsiveness	1	6	1
Elasticity	1	6	1
Security	6	6	1
Availability			
Feasibility (cost/time)			
Reliability			

# **MITIGATING RISK**

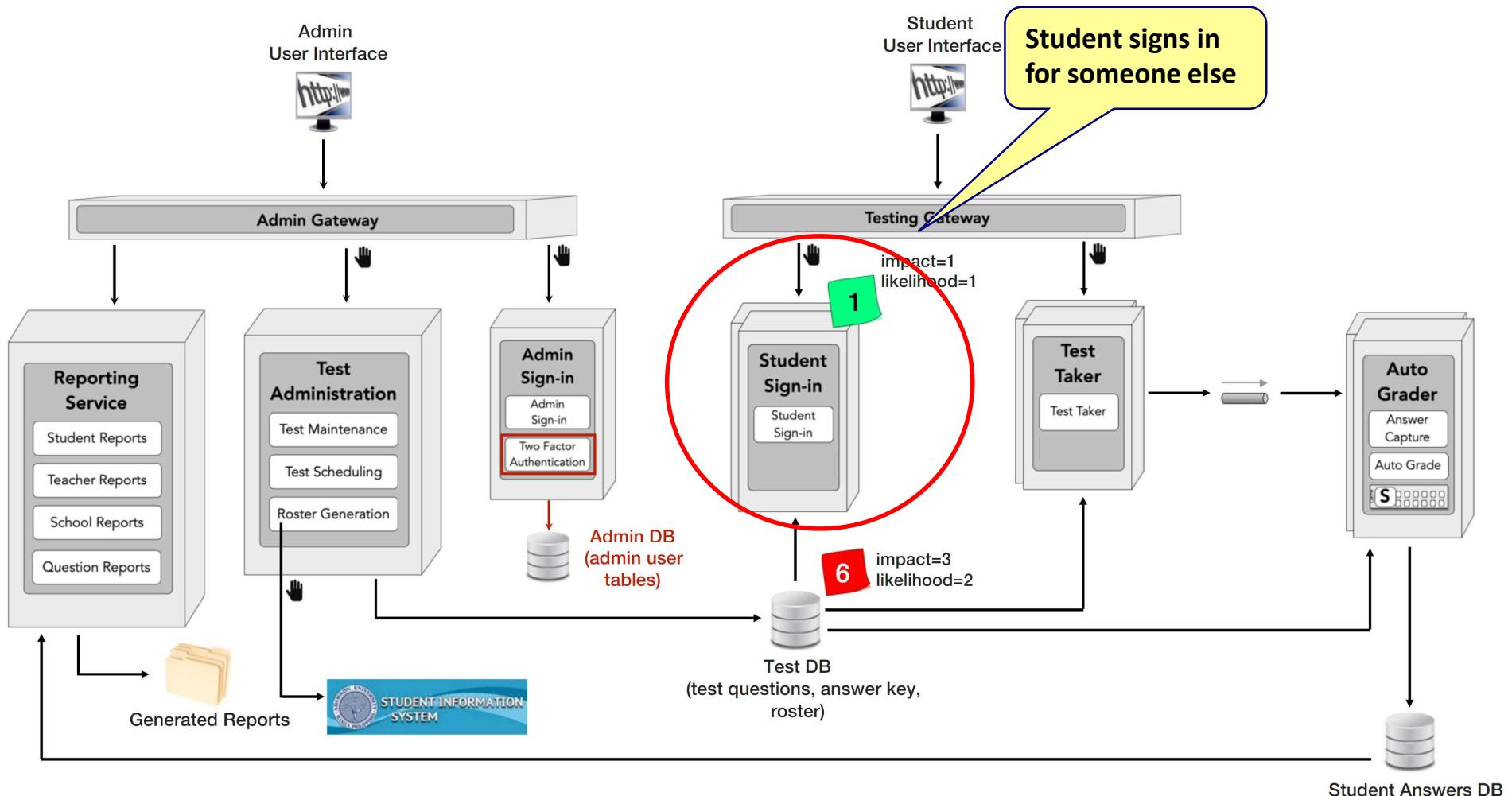
# Security risk



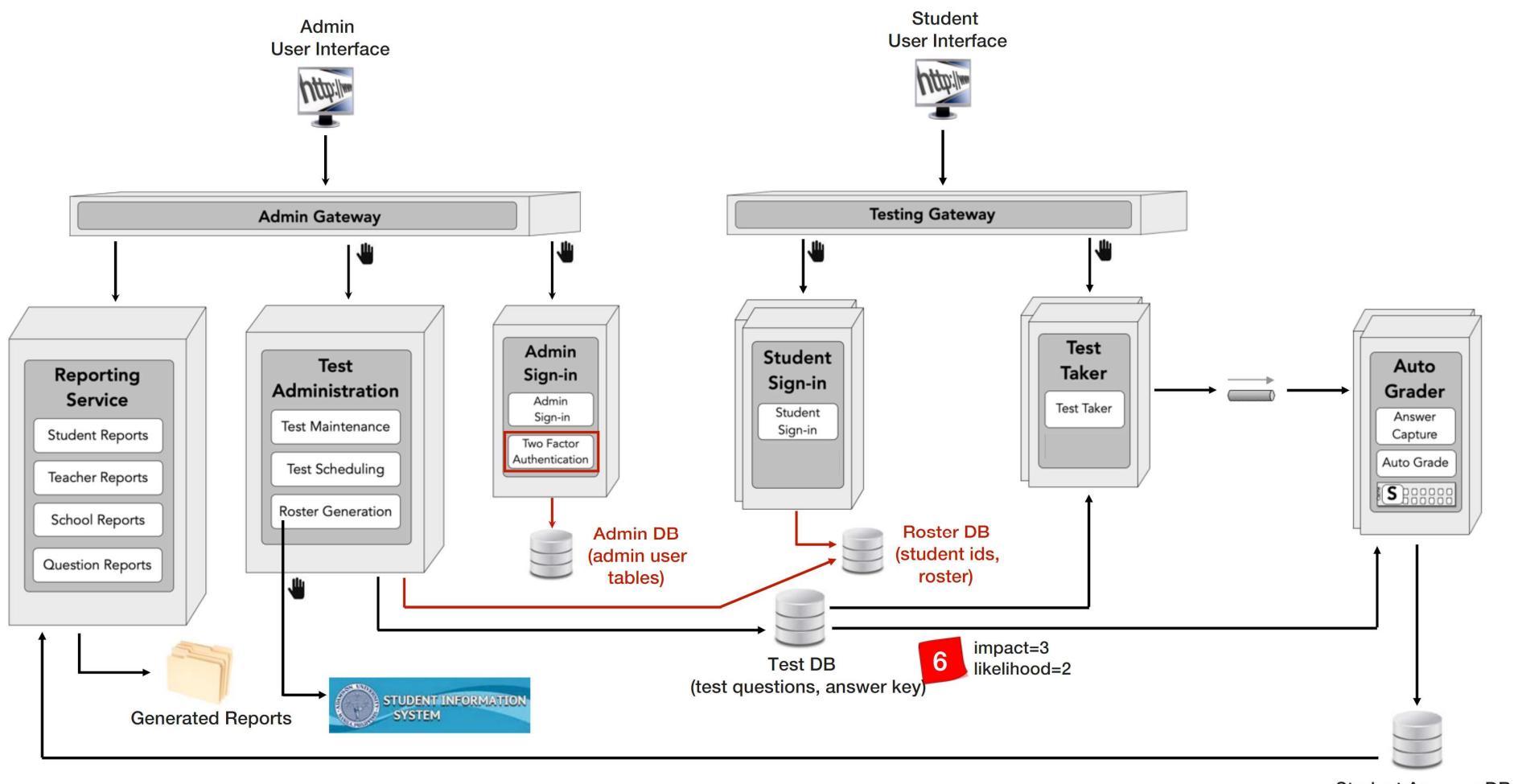
# Security risk mitigation



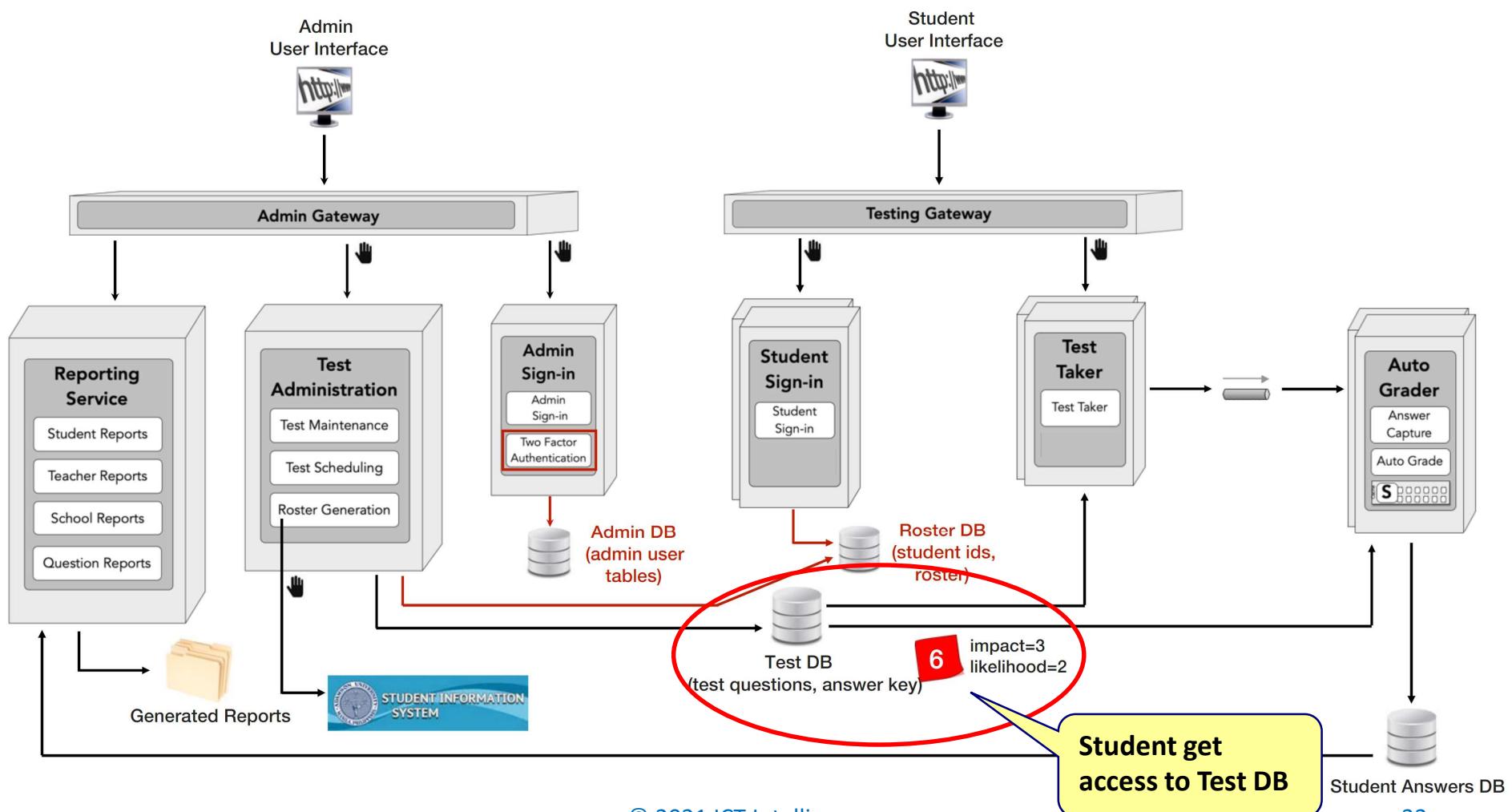
# Security risk mitigation



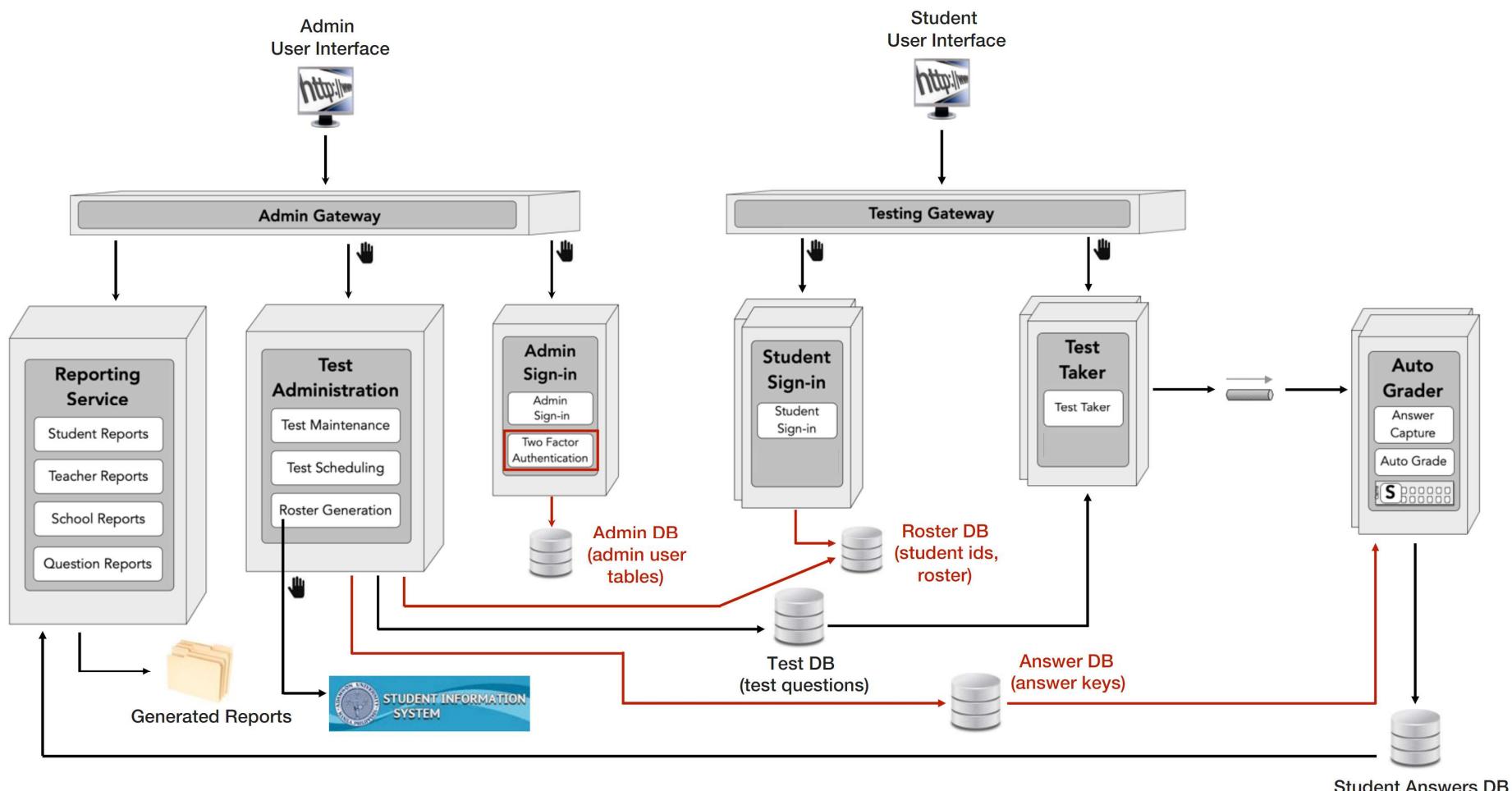
# Security risk mitigation



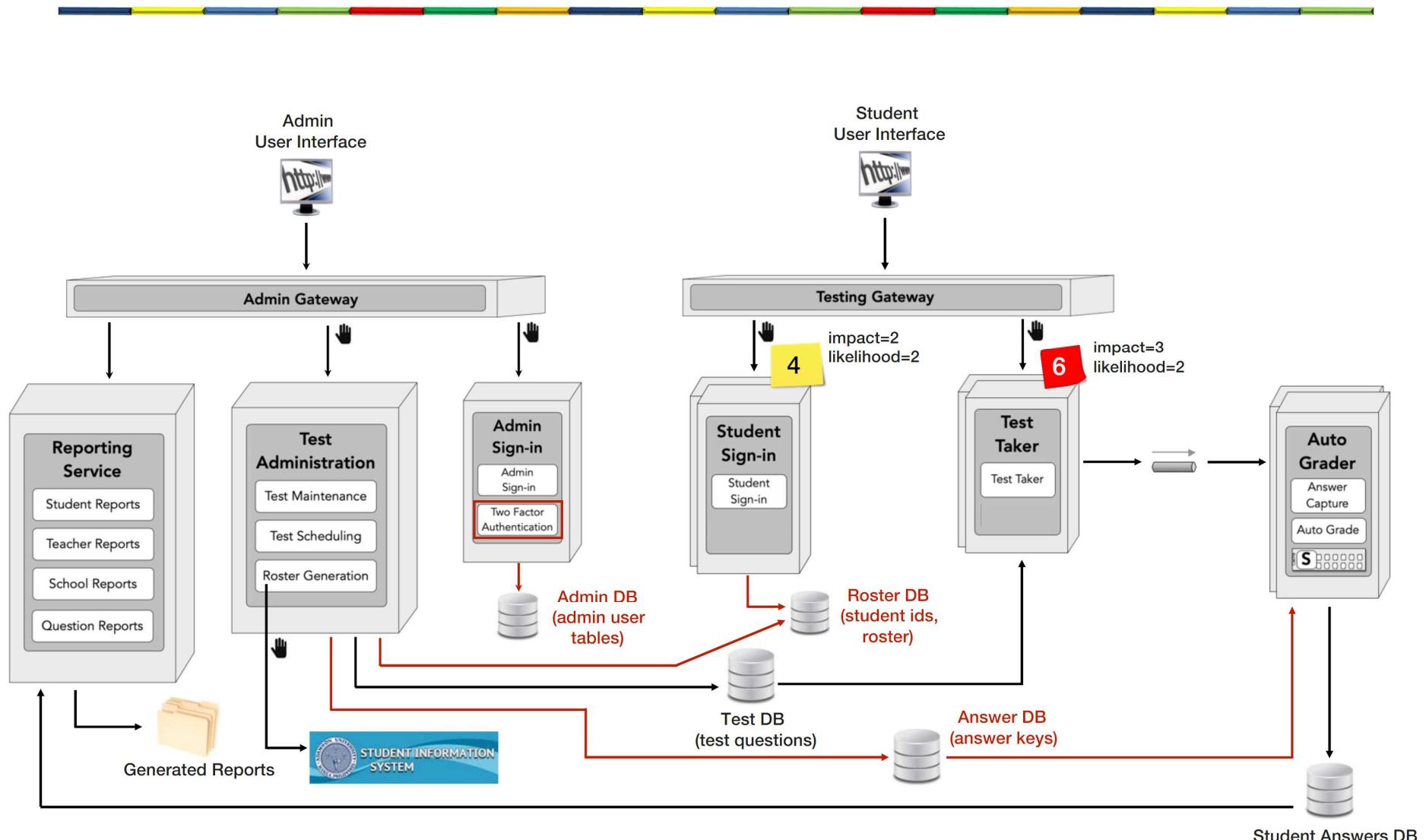
# Security risk mitigation



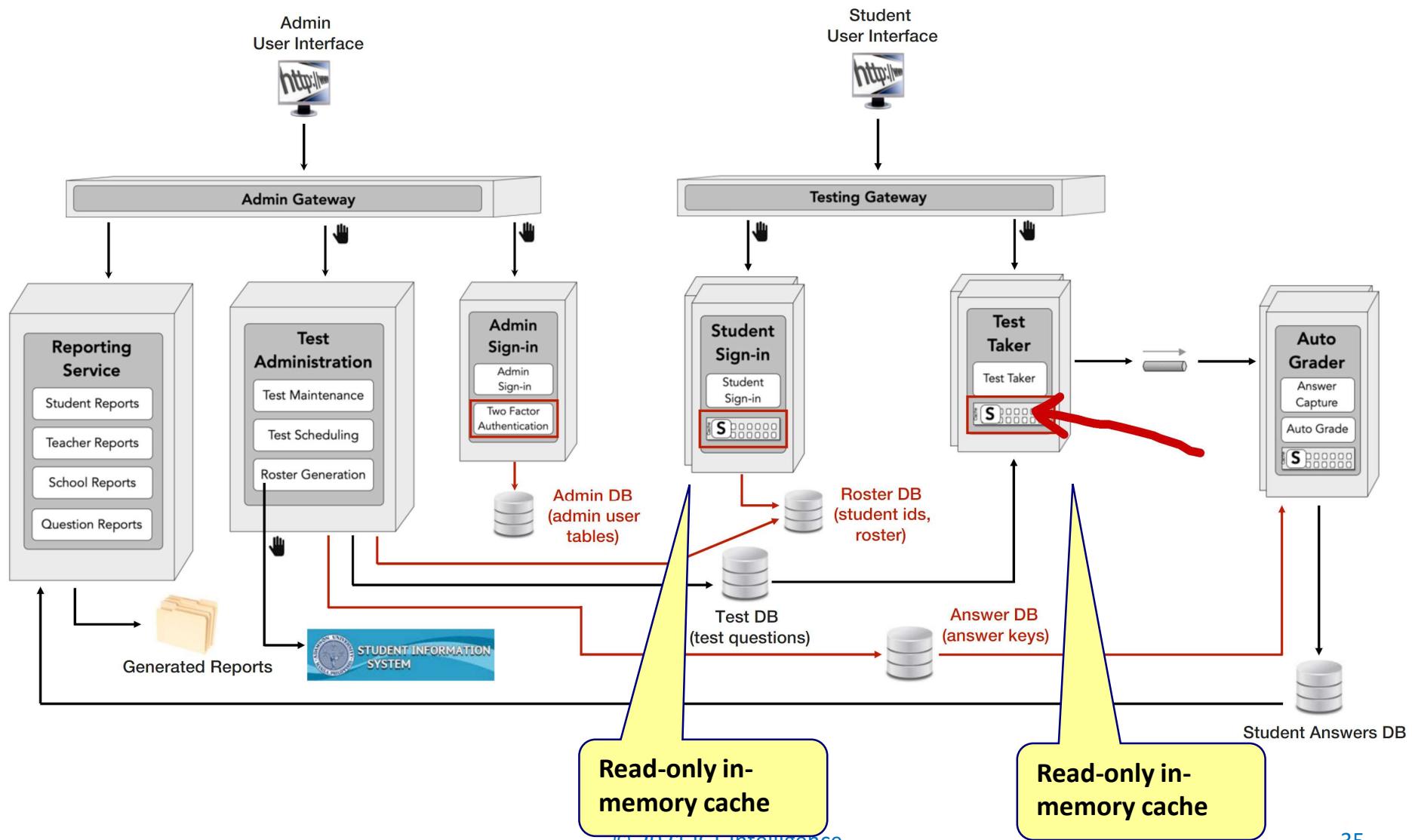
# Security risk mitigation



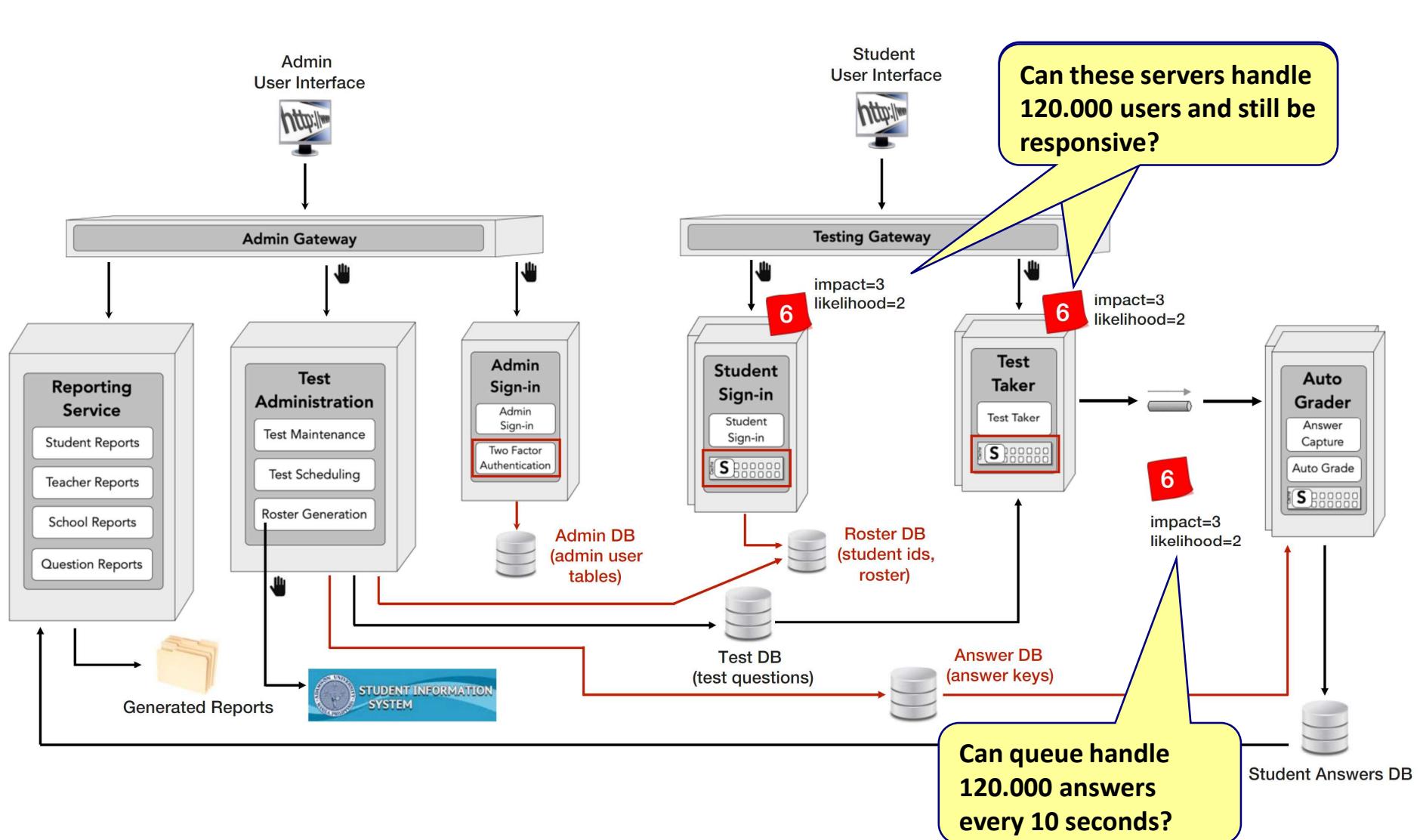
# Responsiveness



# Responsiveness risk mitigation



# Elasticity



# Elasticity risk migration

