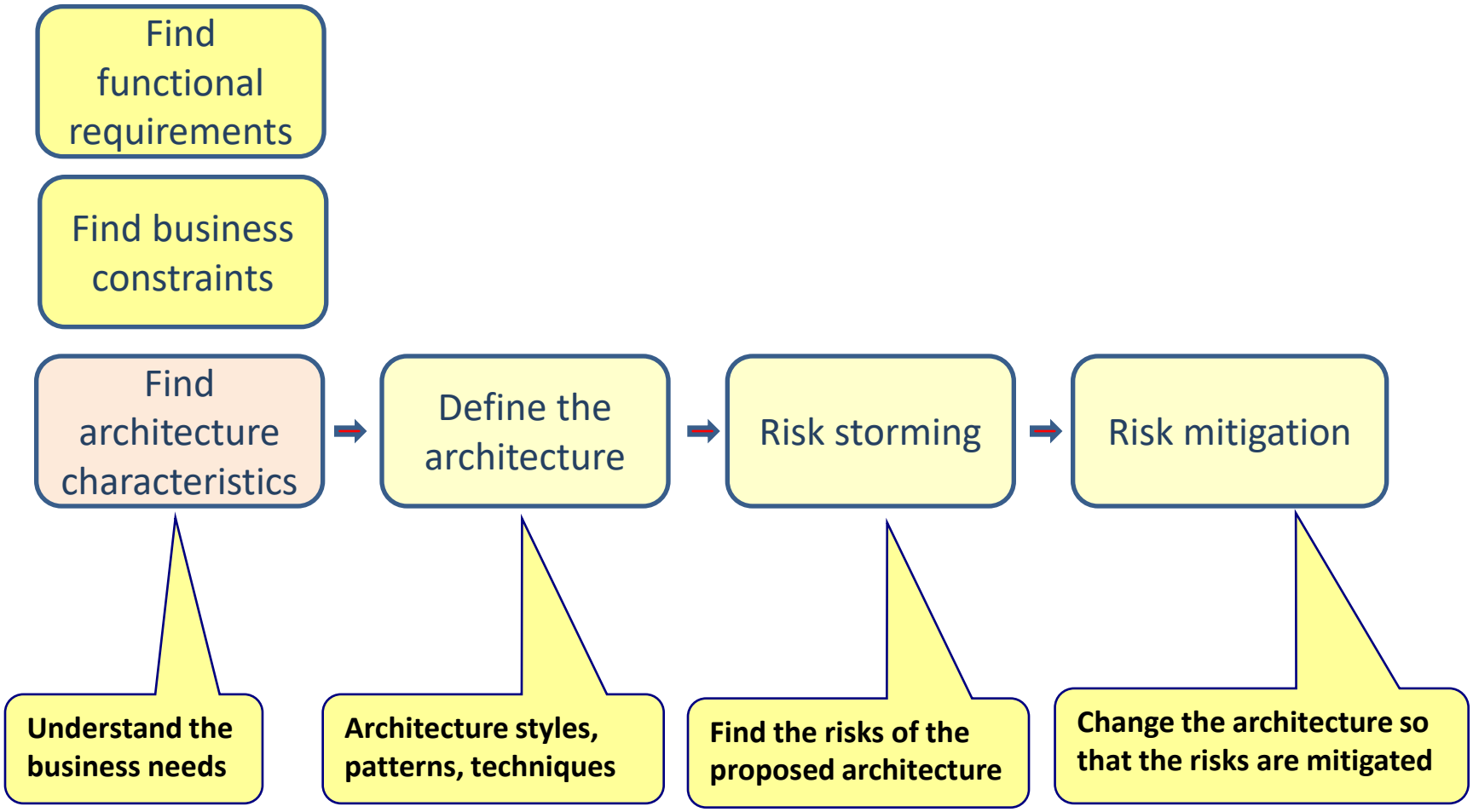
Lesson 13

FINDING THE RIGHT ARCHITECTURE

Finding the right architecture



Functional requirements

- What should the system functionally do?
  - Use cases
  - User stories

As a customer

I can view my account history

so that I know all transactions on my account

Business constraints

- Constraints from the business (or enterprise architecture
  - We do everything in .Net
  - We always use an Oracle database
  - Our maintenance engineers all know Java
  - All applications talk with the Oracle ESB
- Budget
- Deadlines

ARCHITECTURAL CHARACTERISTICS

Architectural characteristics

- Which qualities are available?
- We need to balance the qualities
- A quality itself is not precise enough
- Stakeholders have different interests

Wikipedia software qualities

- accessibility
- accountability
- accuracy
- adaptability
- administrability
- affordability
- agility (see Common subsets below)
- auditability
- autonomy [Erl]
- availability
- compatibility
- composability [Erl]
- configurability
- correctness
- credibility
- customizability
- debuggability
- degradability
- determinability
- demonstrability
- dependability (see Common subsets below)
- deployability
- discoverability [Erl]
- distributability
- durability
- effectiveness
- efficiency
- evolvability
- extensibility
- failure transparency
- fault-tolerance
- fidelity
- flexibility
- inspectability
- installability
- integrity
- interchangeability
- interoperability [Erl]
- learnability
- localizability
- maintainability
- manageability
- mobility
- modifiability
- modularity
- observability
- operability
- orthogonality
- portability
- precision
- predictability
- process capabilities
- producibility
- provability
- recoverability
- relevance
- reliability
- repeatability
- reproducibility
- resilience
- responsiveness
- reusability [Erl]
- robustness
- safety
- scalability
- seamlessness
- self-sustainability
- serviceability (a.k.a. supportability)
- securability (see Common subsets below)
- simplicity
- stability
- standards compliance
- survivability
- sustainability
- tailorability
- testability
- timeliness
- traceability
- transparency
- ubiquity
- understandability
- upgradability
- usability
- vulnerability

SEI quality model

Qualities noticeable at runtime

Performance	**Responsiveness of the system**
Security	**Ability to resist unauthorized usage**
Availability	**Portion of time the system is available**
Functionality	**Ability to do intended work**
Usability	**Learnability, efficiency, satisfaction, error handling, error avoidance**

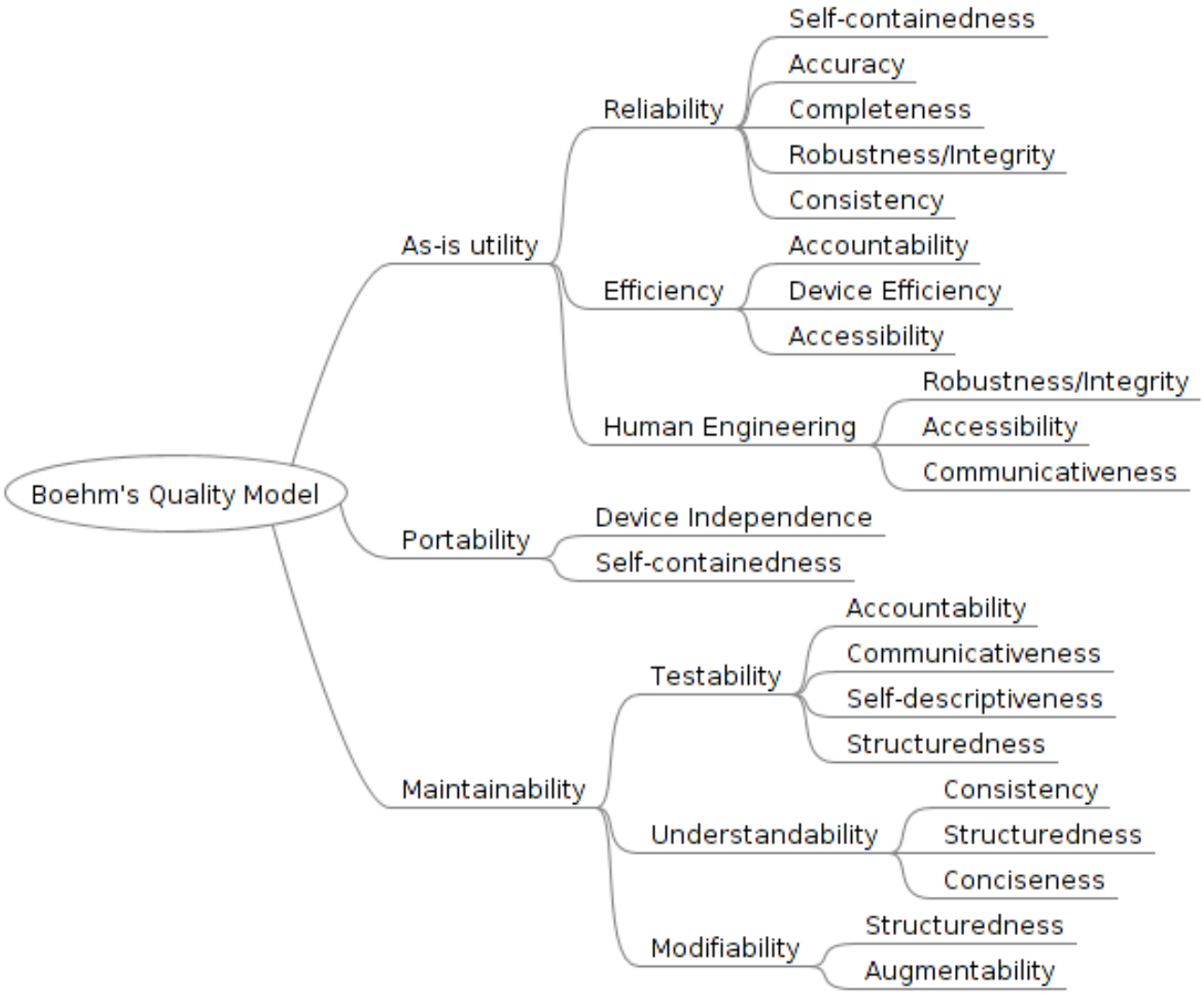
Qualities not noticeable at runtime

Modifiability	**Cost of introducing change**
Portability	**Ability to operate in different computing environments**
Reusability	**Ability to reuse components in different applications**
Integrability	**Ability that components work correctly together**
Testability	**Ability to systematic testing to discover defects**

FURPS model

- **Functionality** - evaluate the feature set and capabilities of the program, the generality of the functions delivered and the security of the overall system
- **Usability** - consider human factors, overall aesthetics, consistency, and documentation
- **Reliability** - measure the frequency and severity of failure, the accuracy of outputs, the ability to recover from failure, and the predictability
- **Performance** - measure the processing speed, response time, resource consumption, throughput and efficiency
- **Supportability** - measure the maintainability, testability, configurability and ease of installation

Boehm



ISO 25010

Organize architectural characteristics

Software development **process**

- **Modifiability** – Ease with which the system can be changed or extended.
- **Testability** – How easily the system's behavior can be tested and verified.
- **Reusability** – Degree to which components can be reused in other systems.
- **Maintainability** – Effort required to identify and fix issues or improve the system.
- **Deployability** – How easily, reliably, and frequently the system can be released and installed into target environments.

Structure of the system

- **Modularity** – Division of the system into independent, cohesive components.
- **Portability** – How easily the system can operate in different environments with minimal changes.
- **Extensibility** – How easily new features can be added to the system without affecting existing functionality.

Operation of the system

- **Performance** – How quickly the system responds and processes data.
- **Availability** – Degree to which the system is operational and accessible when needed.
- **Reliability** – Ability of the system to function correctly over time without failure.
- **Scalability** – Capacity to handle increased load or usage without degradation.
- **Recoverability** – How quickly the system can restore normal operations after a failure or disruption.
- **Robustness** – How well the system continues to function correctly under stress, invalid inputs, or unexpected conditions.

Crosscutting

- **Security** – Protection of the system and its data from unauthorized access, misuse, or attacks.
- **Accessibility** – Ease with which people of all abilities can use and interact with the system.
- **Usability** – How easily and efficiently users can learn, understand, and operate the system.
- **Privacy** – Safeguarding of personal or sensitive information from unauthorized collection, use, or disclosure.

Architectural characteristics

- Which qualities are available?
- We need to balance the qualities
- A quality itself is not precise enough
- Stakeholders have different interests

Balance the qualities

- More security through encription lowers performance
- More scalability through clustering lowers performance
- More scalability through clustering increases the cost

Find the top 5(+/- 2) qualitities

Architectural characteristics

- Which qualities are available?
- We need to balance the qualities
- A quality itself is not precise enough
- Stakeholders have different interests

Quality scenario’s

- A quality on itself has little meaning
- Create scenario’s for the top qualitities
- Make scenario’s measurable
  - The should be able to scale to 1000 concurrent users
  - The system should be available 24/7
  - All user actions should give a response within 3 seconds.
- Prioritize the scenario's
- Write acceptance tests for NFR scenario’s

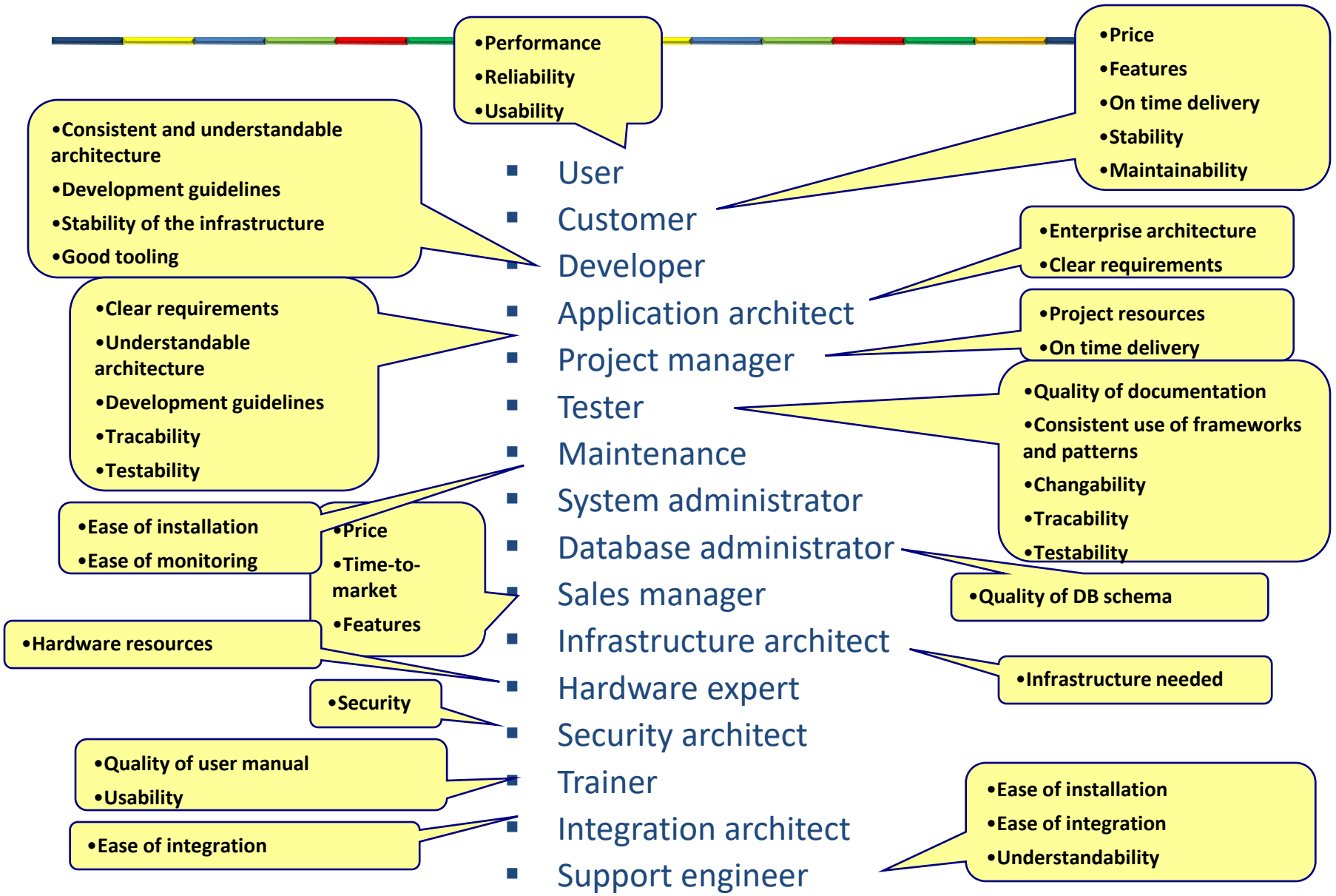
Architectural characteristics

- Which qualities are available?
- We need to balance the qualities
- A quality itself is not precise enough
- Stakeholders have different interests

Stakeholders

- User
- Customer
- Developer
- Application architect
- Project manager
- Tester
- Maintenance
- System administrator
- Database administrator
- Sales
- Infrastructure architect
- Hardware expert
- Security architect
- Trainer
- Integration architect
- Support engineer

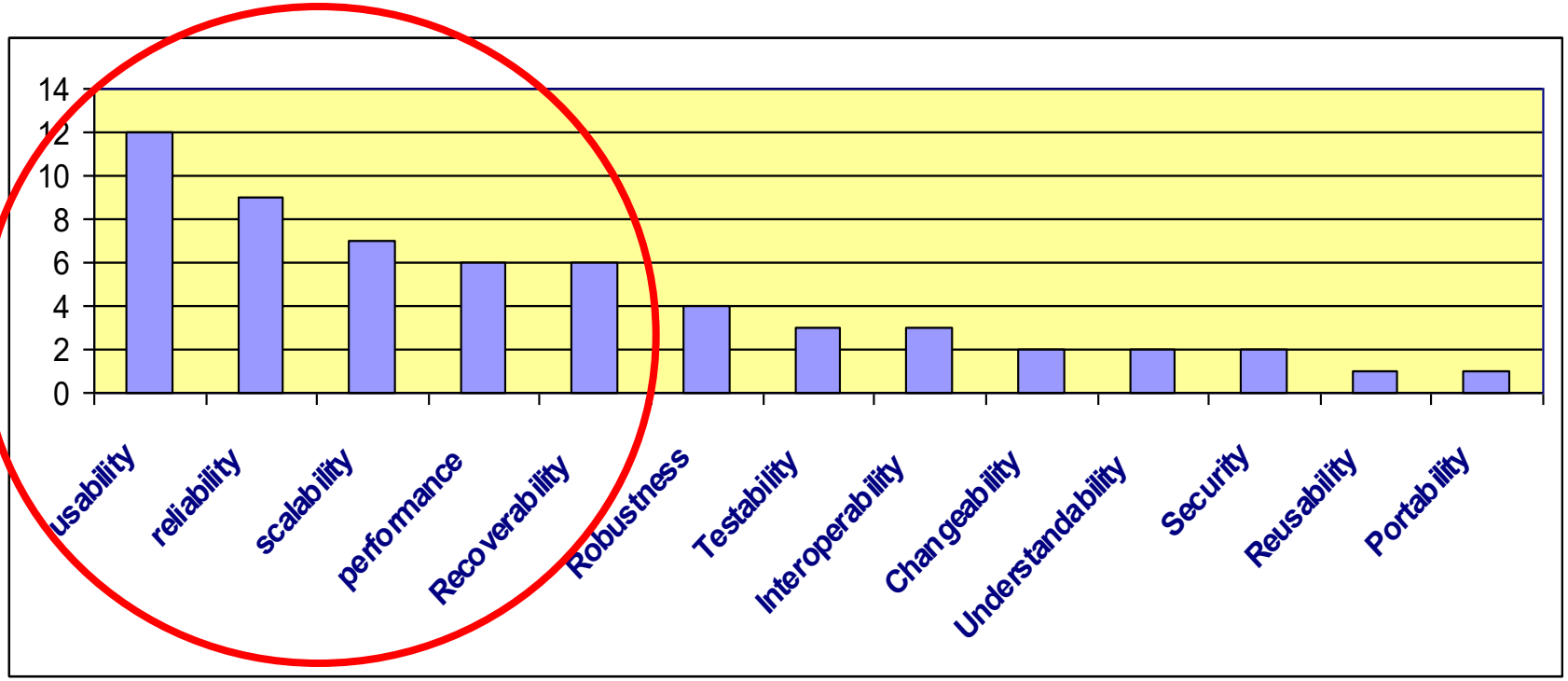
Stakeholders and their interest



Quality workshop

- Goal:
  - Find the prioritized scenario's of the top qualities.
  - Communicate the qualities between stakeholders
- Workshop agenda
  - Explain the vision of the system
  - Explain the different qualitities
  - Everone votes (everyone gets $10 to divide)
  - Discuss the result
  - Vote again
  - Create scenario's for the top qualitites
  - Prioritice the scenario's (vote)

Qualitity workshop result



Some important architectural characteristics

- Performance
- Scalability
- Availability
- Recoverability
- Adaptability
- Maintainability
- Security
- Testability
- Fault-tolerance
- Deployability

Performance

- The responsiveness of the application to perform specific actions in a given time span.

- Scenario’s
  - All actions must respond in 3 seconds
  - Complex actions must respond in 5 seconds

Availability

- The probability that the system is operating properly when it is requested for use
  - Calculated as uptime/total time
- Scenario's
  - The critical part of the system should be available 99.5% of the time
  - The non critical part of the system should be available 98.3% of the time

Availability

Availability %	Downtime per year[note 1]	Downtime per month	Downtime per week	Downtime per day
55.5555555% ("nine fives")	162.33 days	13.53 days	74.92 hours	10.67 hours
90% ("one nine")	**36.53 days**	**73.05 hours**	**16.80 hours**	**2.40 hours**
95% ("one and a half nines")	18.26 days	36.53 hours	8.40 hours	1.20 hours
97%	10.96 days	21.92 hours	5.04 hours	43.20 minutes
98%	7.31 days	14.61 hours	3.36 hours	28.80 minutes
99% ("two nines")	**3.65 days**	**7.31 hours**	**1.68 hours**	**14.40 minutes**
99.5% ("two and a half nines")	1.83 days	3.65 hours	50.40 minutes	7.20 minutes
99.8%	17.53 hours	87.66 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.77 hours**	**43.83 minutes**	**10.08 minutes**	**1.44 minutes**
99.95% ("three and a half nines")	4.38 hours	21.92 minutes	5.04 minutes	43.20 seconds
99.99% ("four nines")	**52.60 minutes**	**4.38 minutes**	**1.01 minutes**	**8.64 seconds**
99.995% ("four and a half nines")	26.30 minutes	2.19 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**26.30 seconds**	**6.05 seconds**	**864.00 milliseconds**
99.9999% ("six nines")	**31.56 seconds**	**2.63 seconds**	**604.80 milliseconds**	**86.40 milliseconds**
99.99999% ("seven nines")	**3.16 seconds**	**262.98 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.58 milliseconds**	**26.30 milliseconds**	**6.05 milliseconds**	**864.00 microseconds**
99.9999999% ("nine nines")	**31.56 milliseconds**	**2.63 milliseconds**	**604.80 microseconds**	**86.40 microseconds**

Recoverability

- How easy can the system recover from failure
- Scenario’s
  - Mean Time To Recovery = 10 hours

Scalability

- The ability to handle an increase in the workload without impacting the performance

- Scenario's
  - The system should scale up to 100.000 users in a year
  - The system should be able to handle 500 concurrent users

Adaptability

- Easy of which a system can be changed

- Scenario's
  - It should be easy to change the database
  - It should be easy to support different clients (web, mobile apps, etc)

Maintainability

- The ability of any application to go through modifications and updates with a degree of ease.
- Scenario’s
  - The system should be highly configurable
  - The system should have 80% test coverage
  - All errors should be logged

Security

- System's ability to resist unauthorized usage while still providing its services to legitimate users
- 3 aspects
  - Authentication: are you who you say you are?
  - Authorization: what are you allowed to do?
  - Secrecy: encrypt the data
- Scenario’s
  - Only authorized users may access the system
  - We always use 2 factor authentication
  - All secret data that is sent to other systems should be encrypted

Testability

- The ease with which software can be made to demonstrate its faults through testing.

- Scenario's
  - We use automated tests
  - We need 80% test coverage

Fault-tolerance

- Ability to continue to operate properly in the event of failure

- Scenario's
  - The system should be fault tolerant for database failure
  - The system should be fault tolerant for network failure

Deployability

- Ease of installing the software

- Scenario’s
  - The system should run on-premise and in the cloud with minimal changes in the configuration
  - There is an automatic installation script

Other important architectural characteristics

- Performance
- Scalability
- Availability
- Recoverability
- Elasticity
- Adaptability
- Maintainability
- Security
- Testability
- Fault-tolerance
- Deployability
- Cost
- Time to market
- Simplicity
- Number of agile teams
- Try out new technology

Auction system

- An auction company wants to take their auctions online to a nationwide scale. Customers choose the auction to participate in, wait until the auction begins, then bid during the live auction as if they were there in the room, with the auctioneer.
- **Users**: scale up to hundreds of participants (per auction), potentially up to thousands of participants, and as many simultaneous auctions as possible
- **Requirements**:
  - bidders can see a live video stream of the auction and see all bids as they occur
  - auctions must be as real-time as possible
  - both online and live bids must be received in the order in which they are placed
  - bidders register with credit card; system automatically charges card if bidder wins
- **• Additional Context:**
  - auction company is expanding aggressively by merging with smaller competitors
  - if nationwide auction is a success, replicate the model overseas
  - budget is not constrained--this is a strategic direction
  - company just exited a lawsuit where they settled a suit alleging fraud

Auction system

- An auction company wants to take their auctions online to a nationwide scale. Customers choose the auction to participate in, wait until the auction begins, then bid during the live auction as if they were there in the room, with the auctioneer.
- **Users**: scale up to hundreds of participants (per auction), potentially up to thousands of participants, and as many simultaneous auctions as possible
- **Requirements**:
  - bidders can see a live video stream of the auction and see all bids as they occur
  - auctions must be as real-time as possible
  - both online and live bids must be received in the order in which they are placed
  - bidders register with credit card; system automatically charges card if bidder wins
- • **Additional Context:**
  - auction company is expanding aggressively by merging with smaller competitors
  - if nationwide auction is a success, replicate the model overseas
  - budget is not constrained--this is a strategic direction
  - company just exited a lawsuit where they settled a suit alleging fraud

availability performance

Auction system

- An auction company wants to take their auctions online to a nationwide scale. Customers choose the auction to participate in, wait until the auction begins, then bid during the live auction as if they were there in the room, with the auctioneer.
- **Users**: scale up to hundreds of participants (per auction), potentially up to thousands of participants, and as many simultaneous auctions as possible
- **Requirements**:
  - bidders can see a live video stream of the auction and see all bids as they occur
  - auctions must be as real-time as possible
  - both online and live bids must be received in the order in which they are placed
  - bidders register with credit card; system automatically charges card if bidder wins
- • **Additional Context:**
  - auction company is expanding aggressively by merging with smaller competitors
  - if nationwide auction is a success, replicate the model overseas
  - budget is not constrained--this is a strategic direction
  - company just exited a lawsuit where they settled a suit alleging fraud

availability performance scalability

Auction system

- An auction company wants to take their auctions online to a nationwide scale. Customers choose the auction to participate in, wait until the auction begins, then bid during the live auction as if they were there in the room, with the auctioneer.
- **Users**: scale up to hundreds of participants (per auction), potentially up to thousands of participants, and as many simultaneous auctions as possible
- **Requirements**:
  - bidders can see a live video stream of the auction and see all bids as they occur
  - auctions must be as real-time as possible
  - both online and live bids must be received in the order in which they are placed
  - bidders register with credit card; system automatically charges card if bidder wins
- • **Additional Context:**
  - auction company is expanding aggressively by merging with smaller competitors
  - if nationwide auction is a success, replicate the model overseas
  - budget is not constrained--this is a strategic direction
  - company just exited a lawsuit where they settled a suit alleging fraud

availability performance scalability elasticity

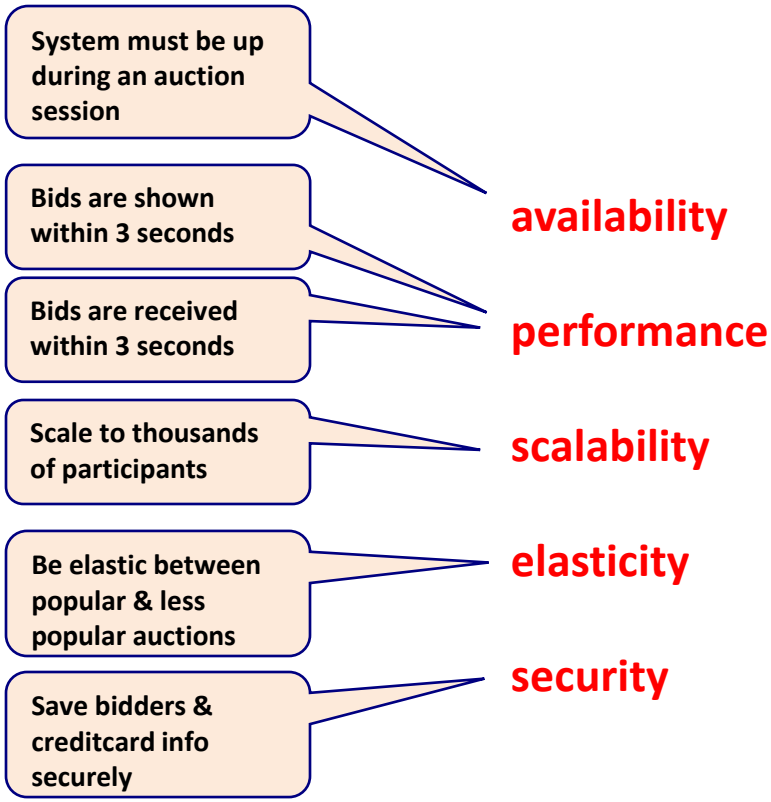
Auction system

- An auction company wants to take their auctions online to a nationwide scale. Customers choose the auction to participate in, wait until the auction begins, then bid during the live auction as if they were there in the room, with the auctioneer.
- **Users**: scale up to hundreds of participants (per auction), potentially up to thousands of participants, and as many simultaneous auctions as possible
- **Requirements**:
  - bidders can see a live video stream of the auction and see all bids as they occur
  - auctions must be as real-time as possible
  - both online and live bids must be received in the order in which they are placed
  - bidders register with credit card; system automatically charges card if bidder wins
- • **Additional Context:**
  - auction company is expanding aggressively by merging with smaller competitors
  - if nationwide auction is a success, replicate the model overseas
  - budget is not constrained--this is a strategic direction
  - company just exited a lawsuit where they settled a suit alleging fraud

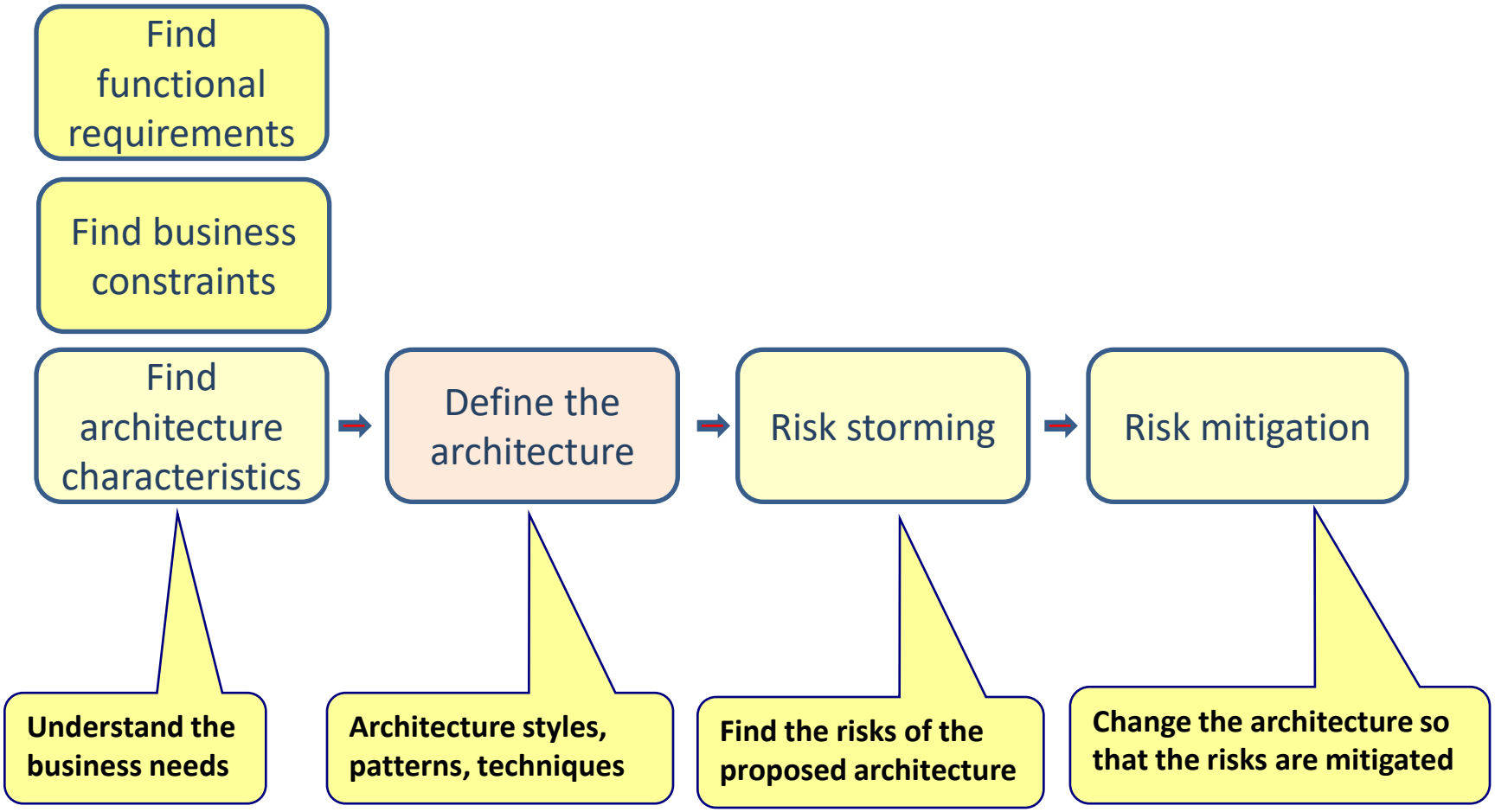
Availability performance scalability elasticity security

Do not choose more than 7 architectural characteristics

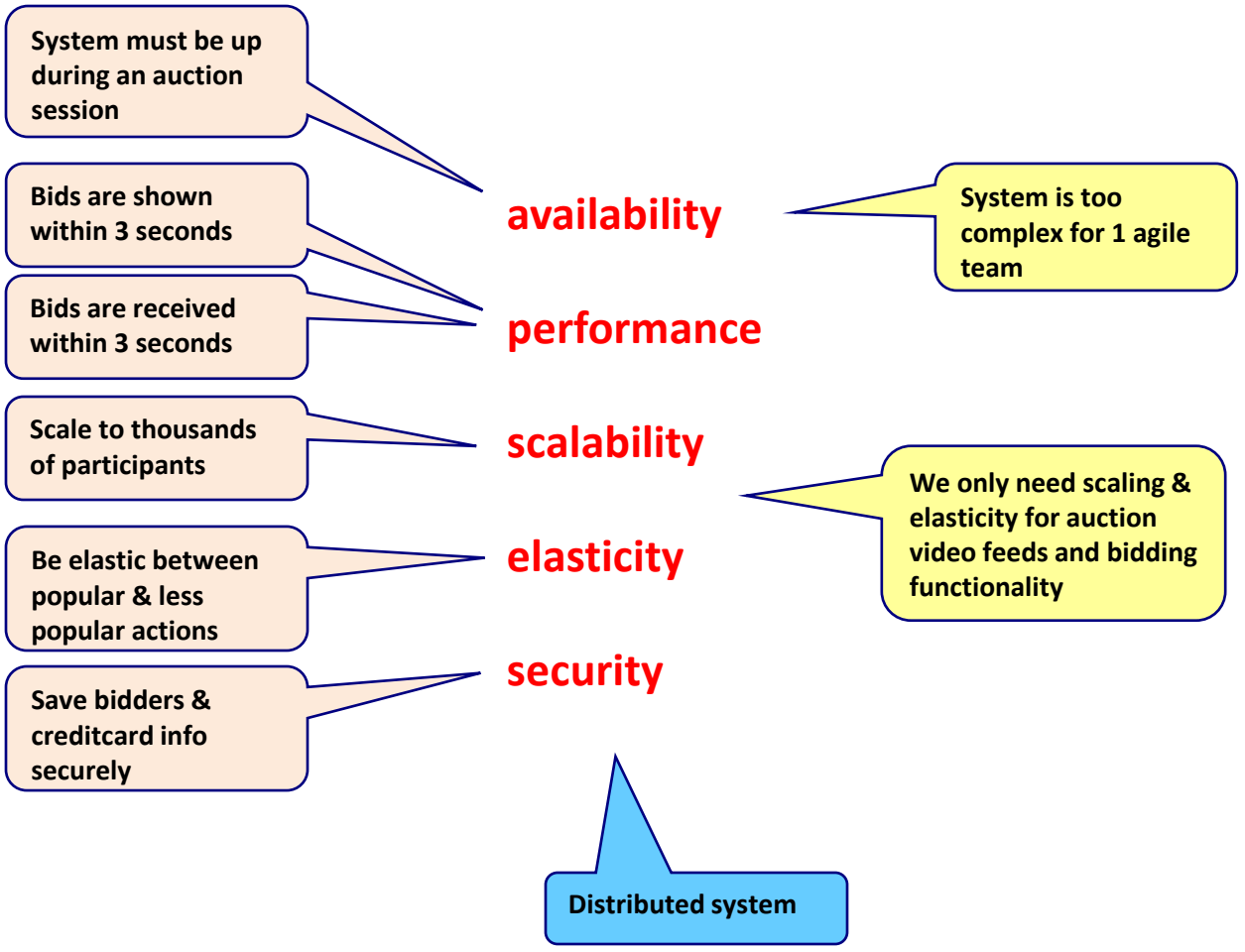
Auction system



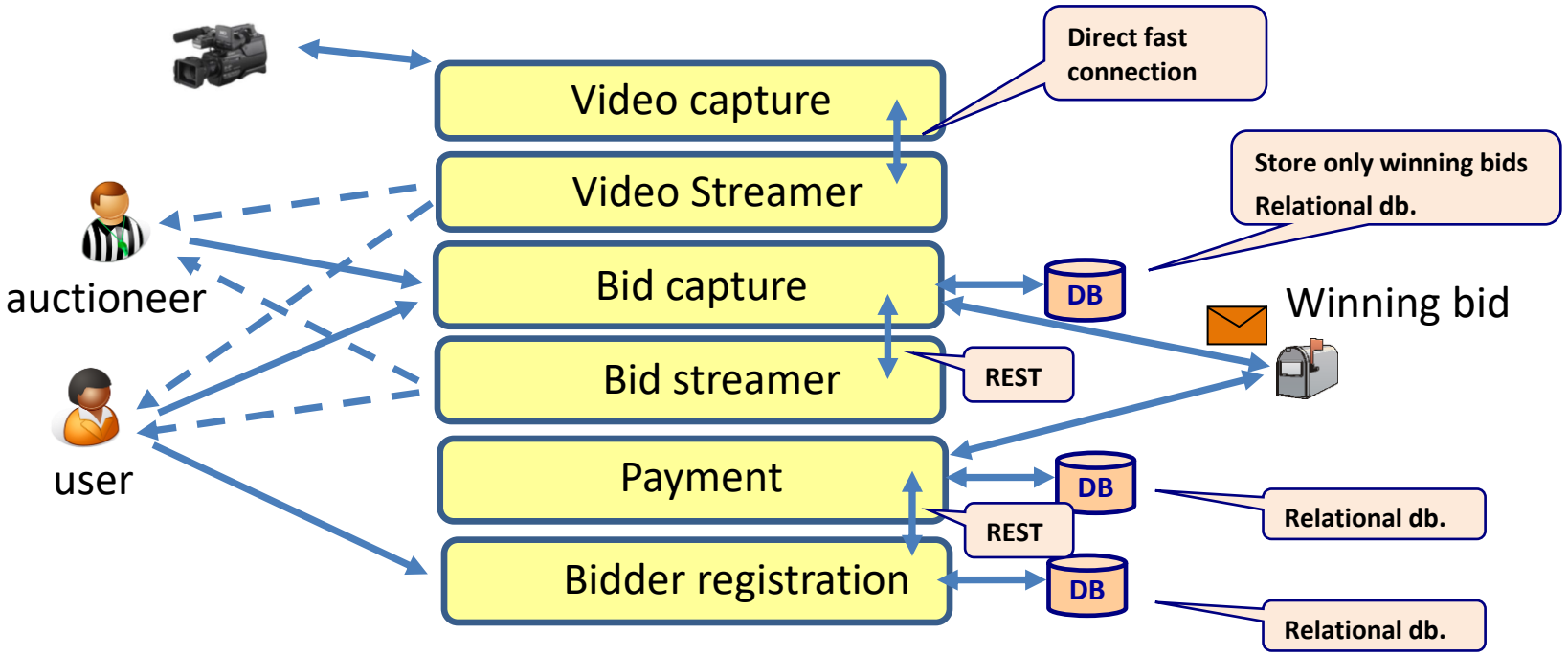
Finding the right architecture



Auction system

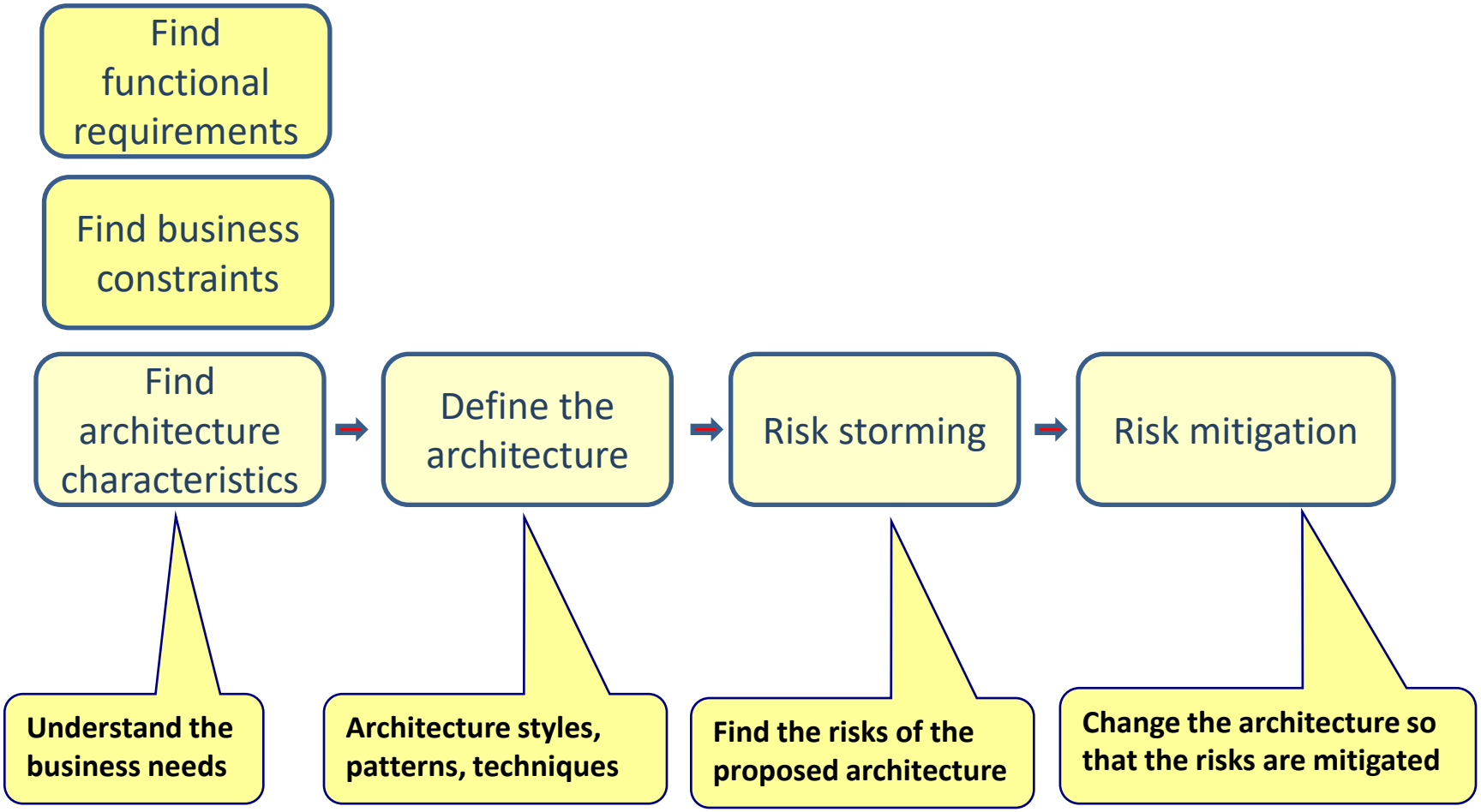


Auction system



Availability reliability performance scalability elasticity security

Software architecture



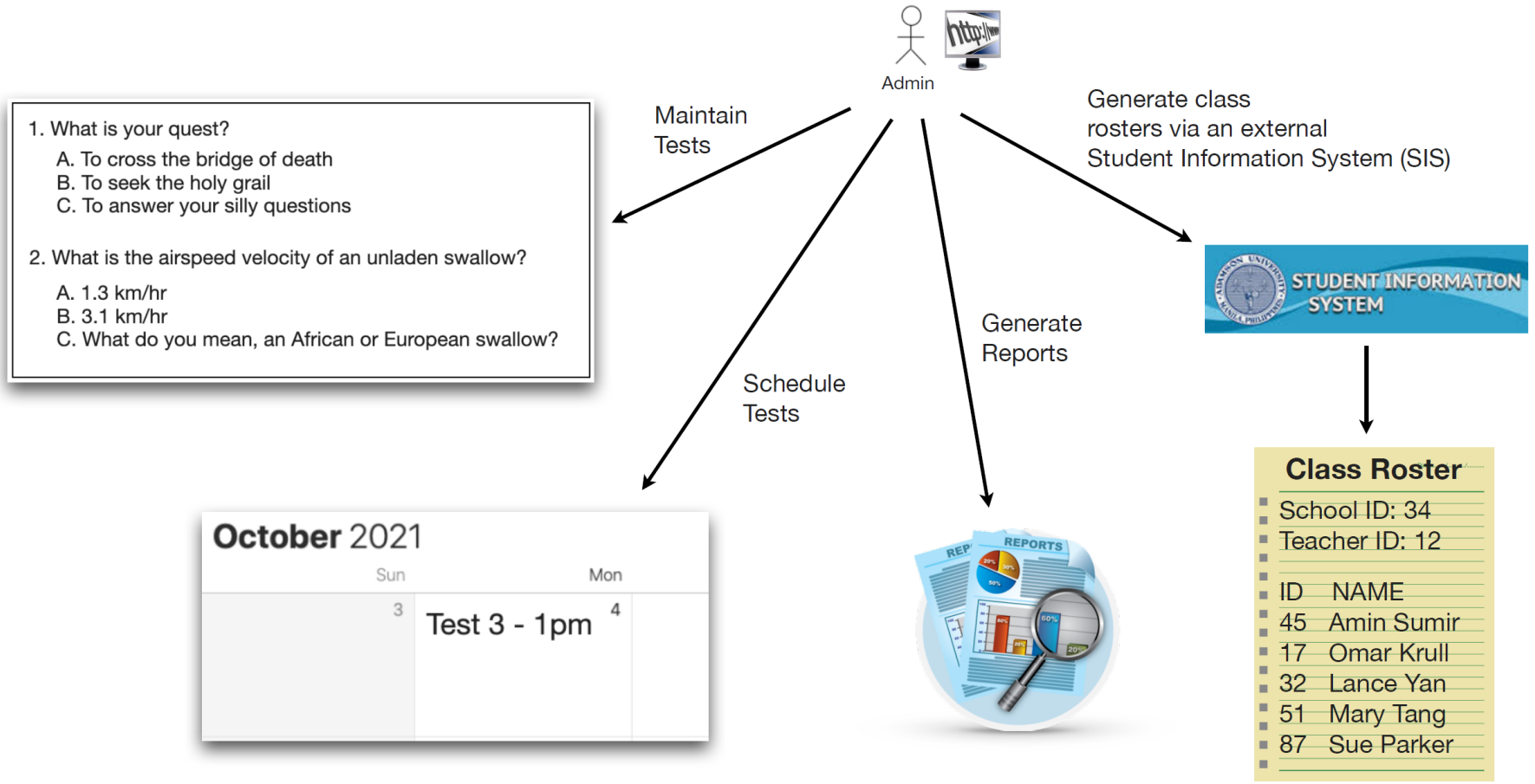
Standardized testing system

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

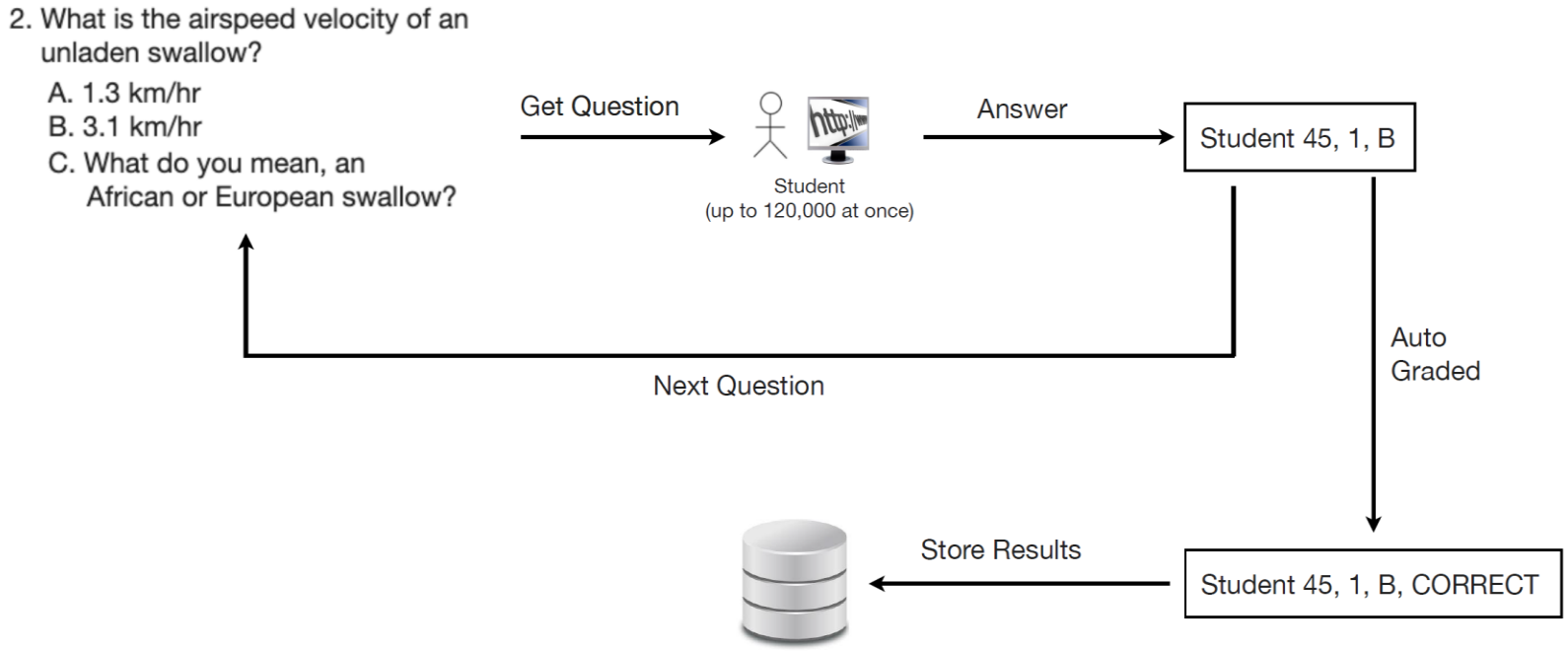
- **Requirements**:
  - Students will use a new web-based (and backend) testing application to take the test from their school based on their assigned teacher. Questions are presented to the student, they submit the answer (true/false or multiple choice), and get the next question. The student answers 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.)

Performance Availability Reliability

Elasticity

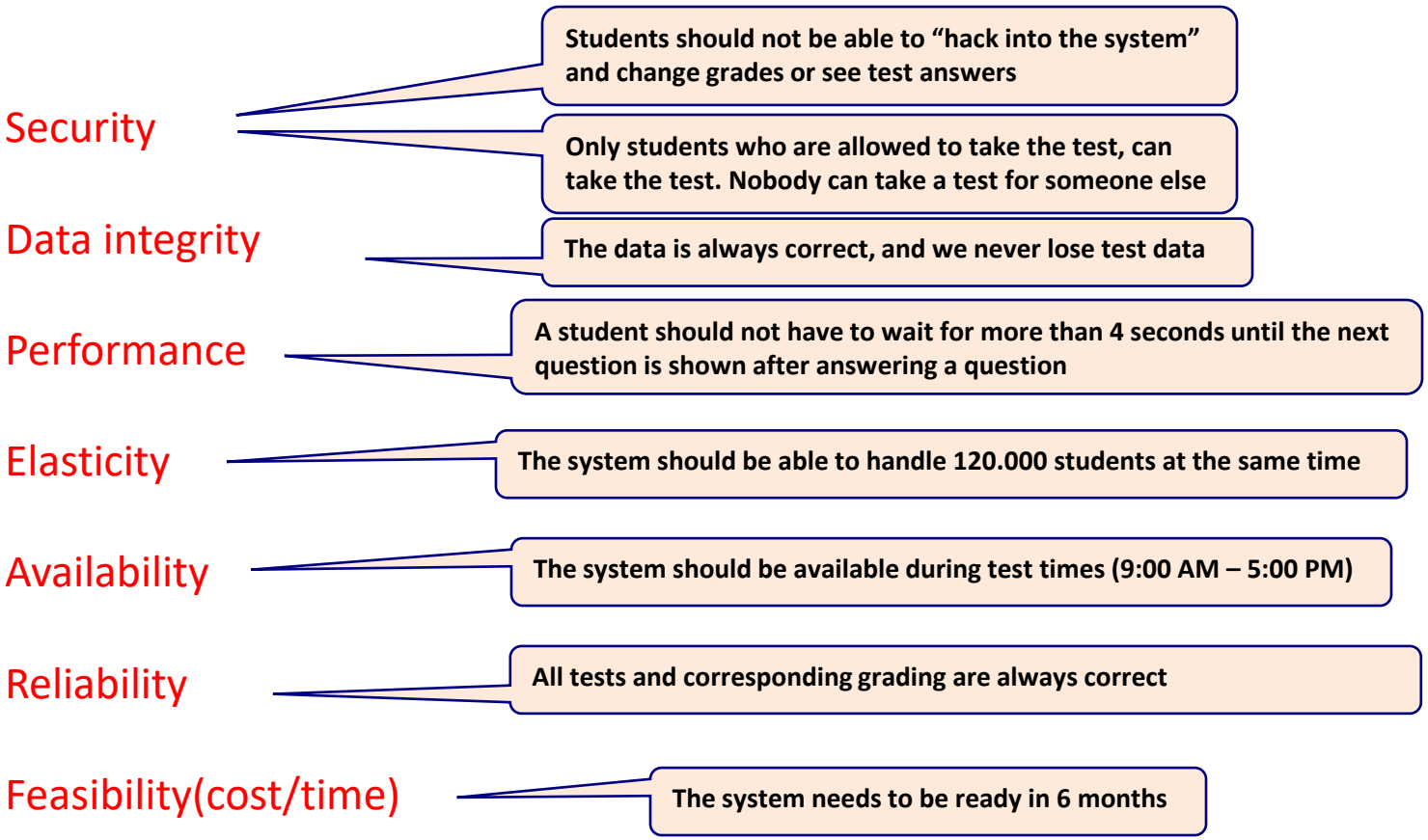
- **Additional info from the education department:**
  - "It is ***absolutely imperative*** that no student answers are lost in the event of a system crash."
  - "We need the new system before the start of the next term (6 months)."
  - "It is vital that the test answers be protected from prying eyes." "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."

Data integrity

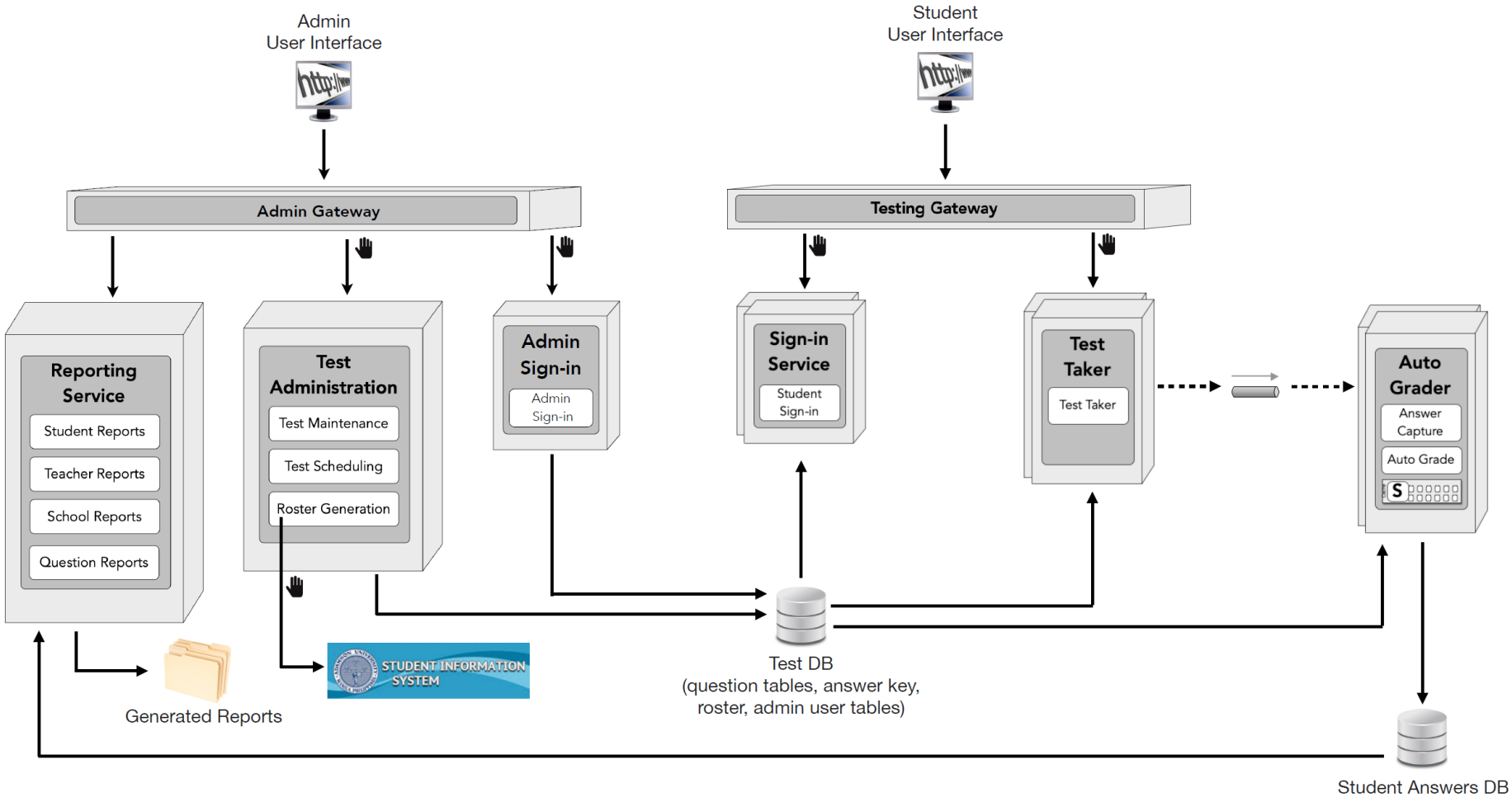
Feasibility(cost/time)

Security

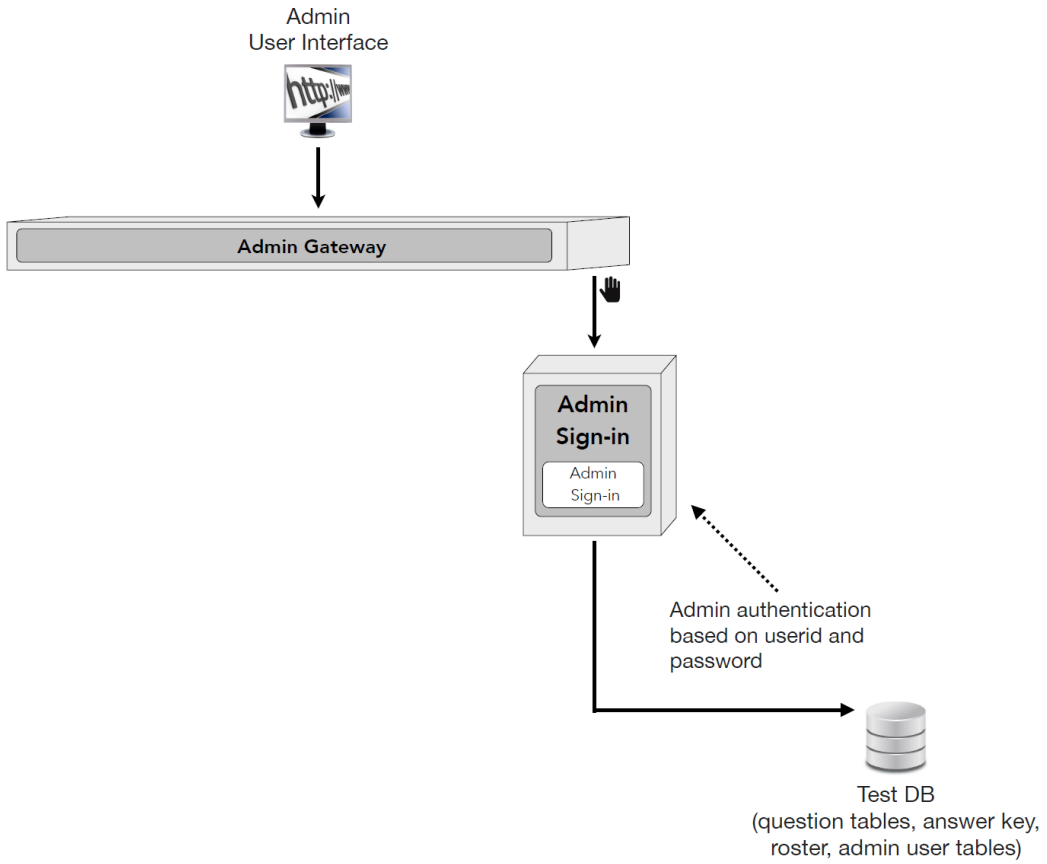
Architectural characteristics



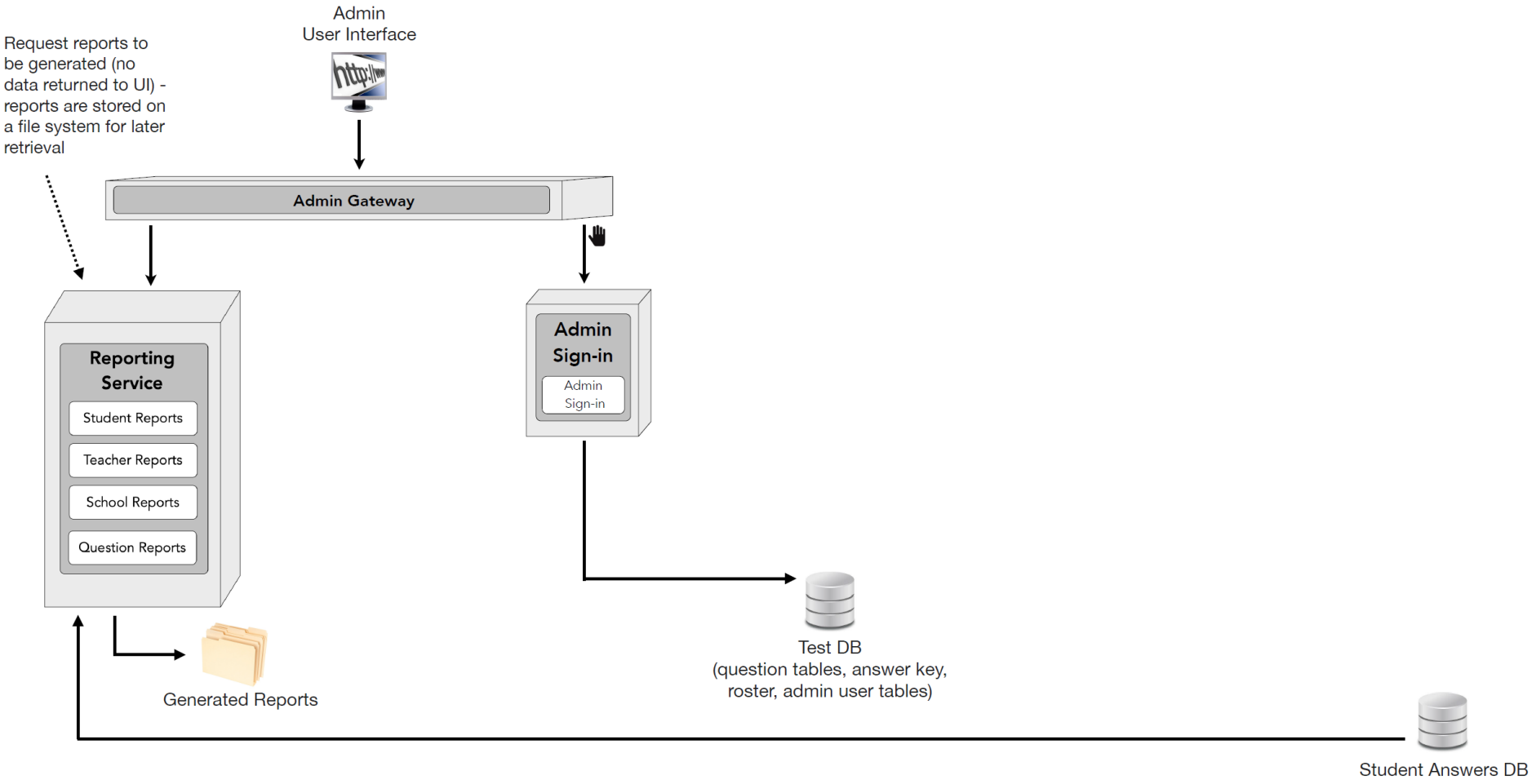
Proposed architecture



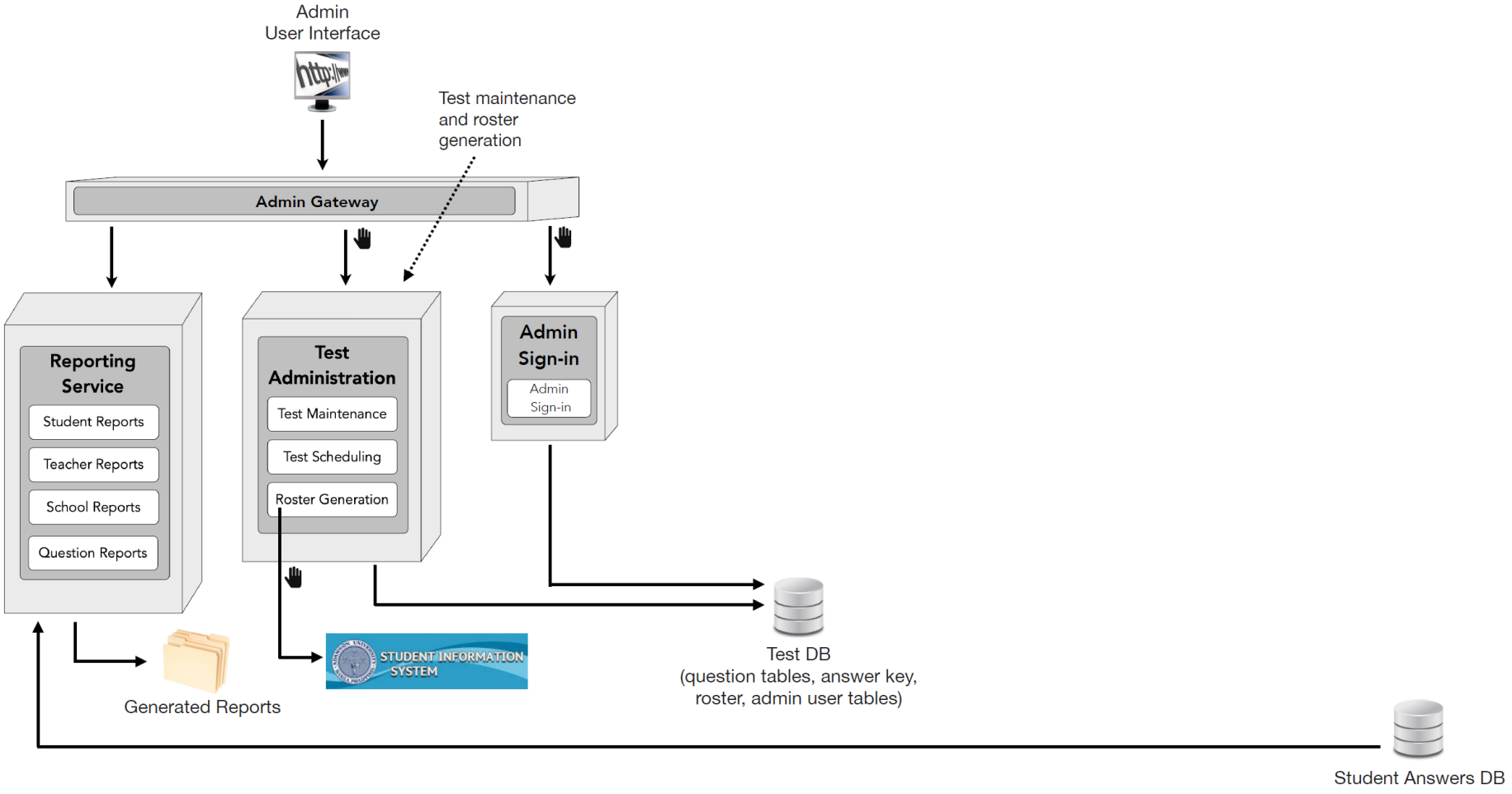
Admin sign-in



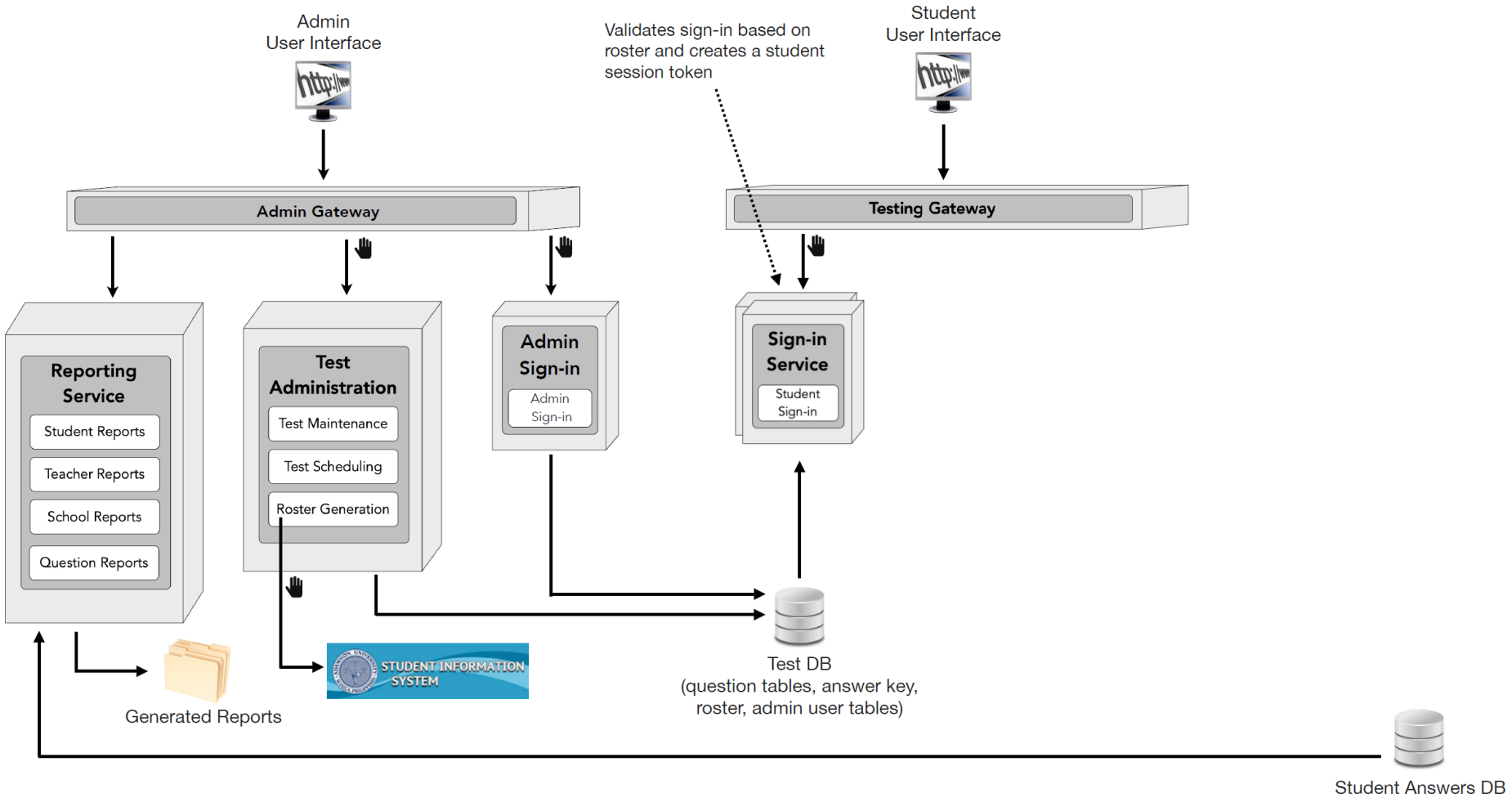
Reporting



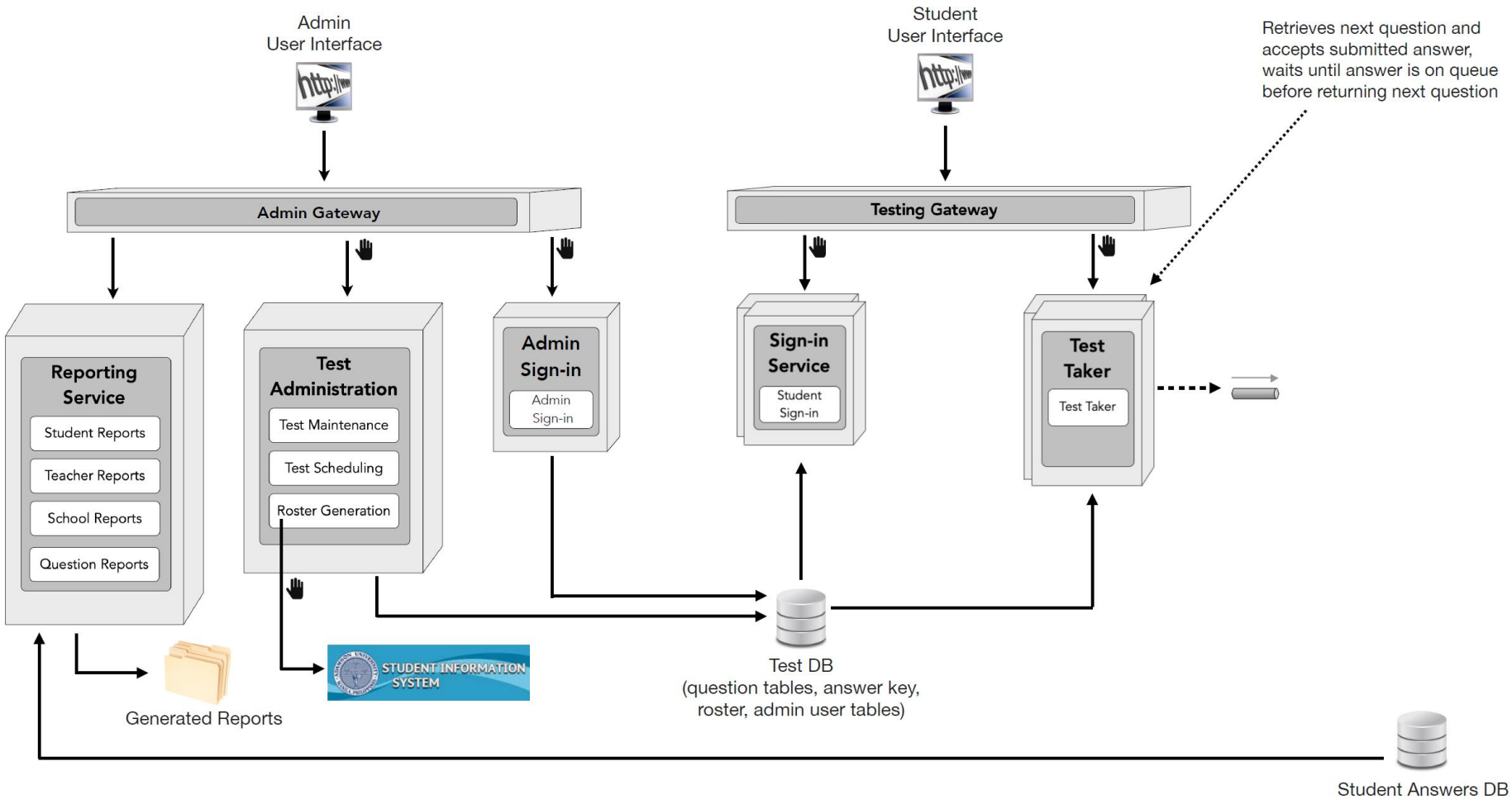
Test administration



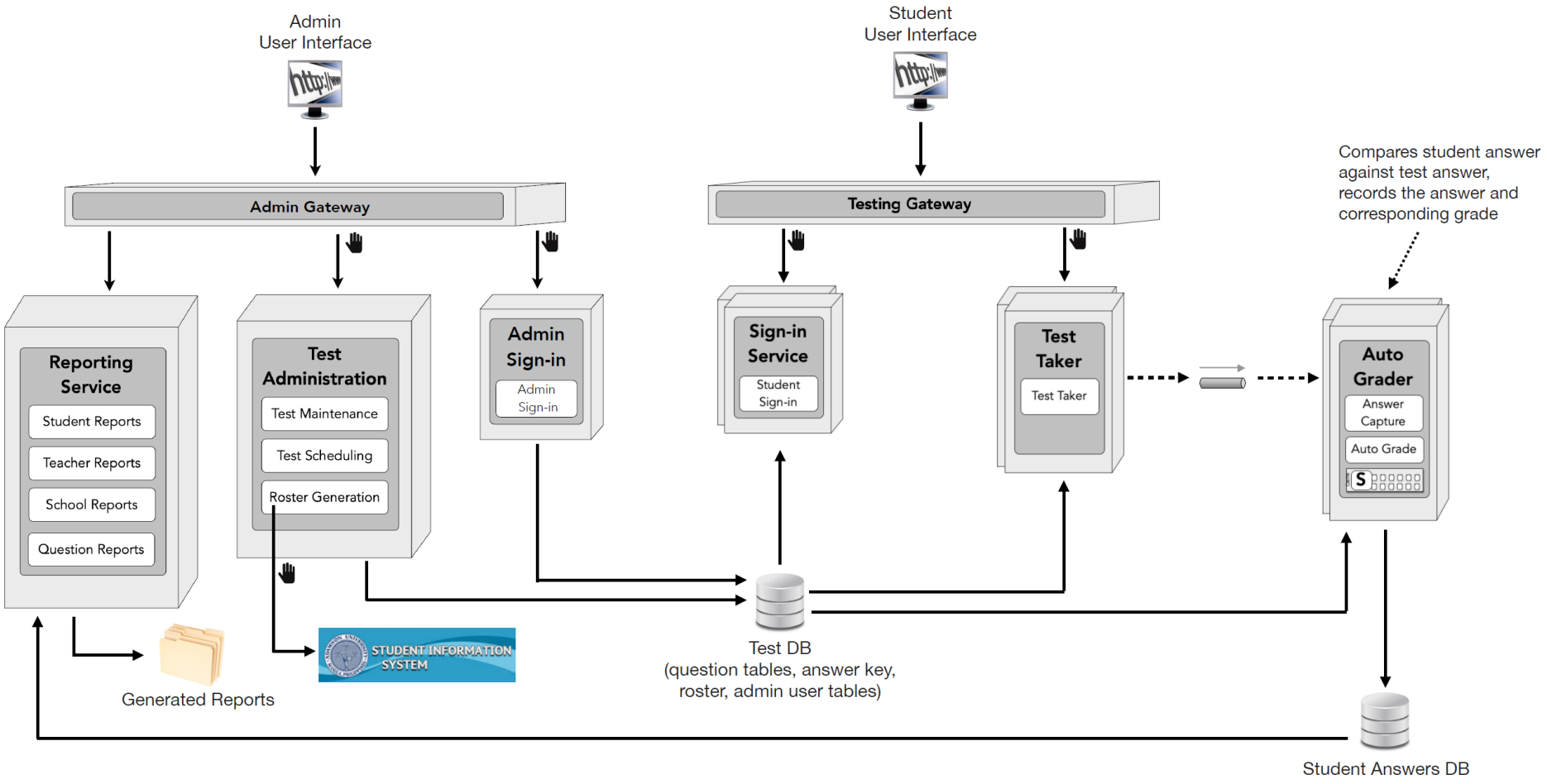
Student sign-in



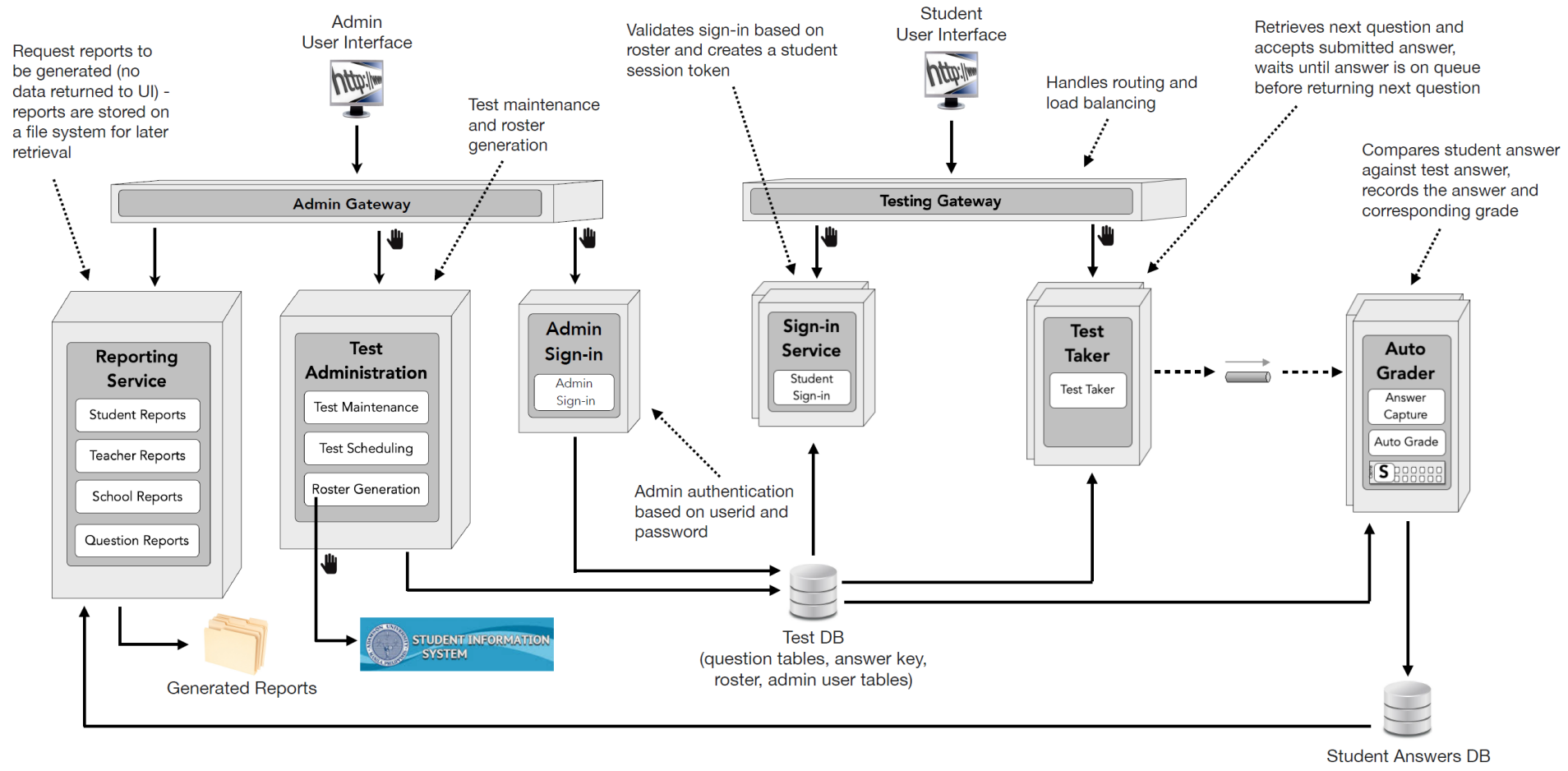
Students take the test



Automated grading

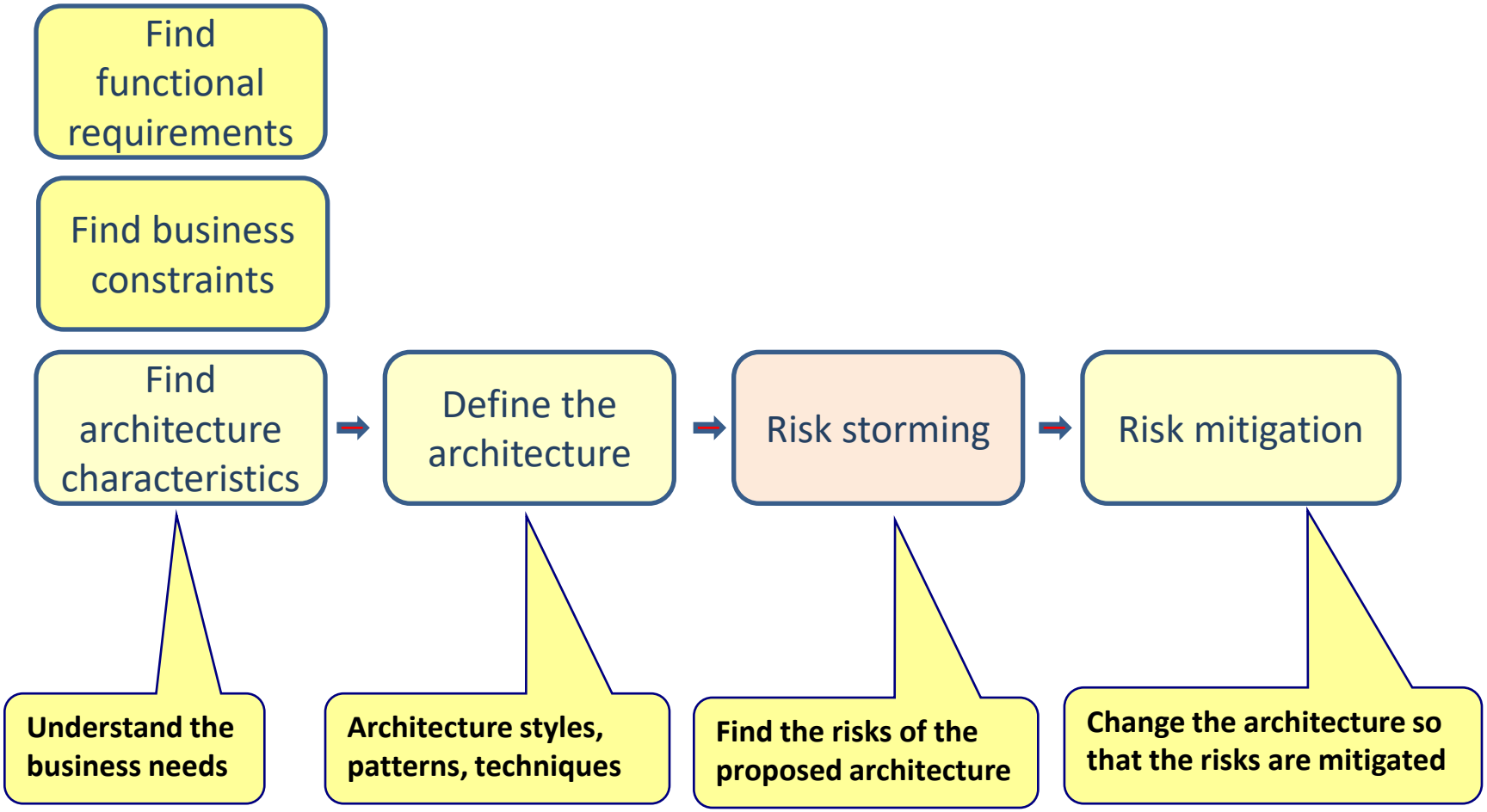


Proposed architecture



RISK STORMING

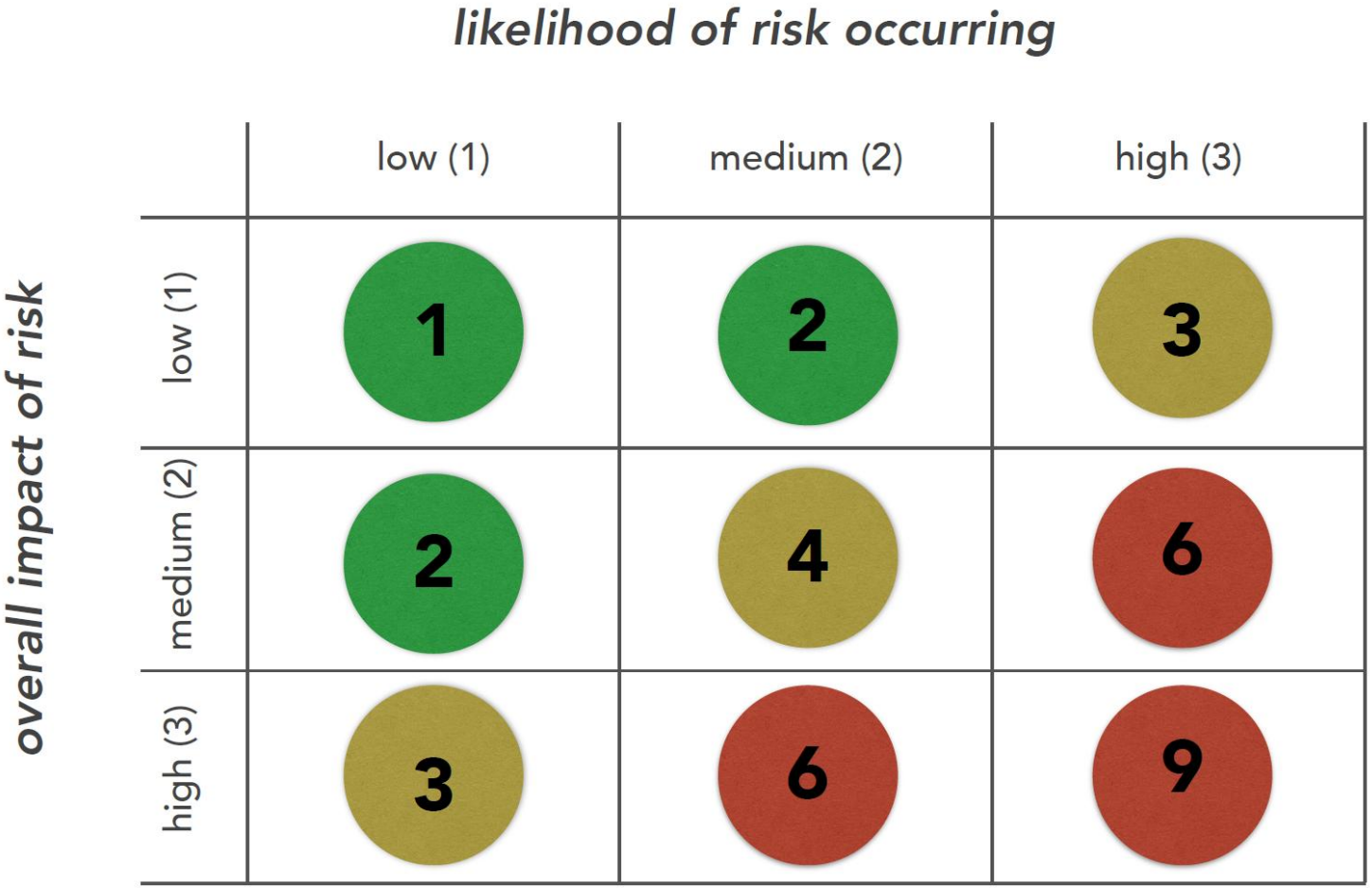
Software architecture



Risk storming

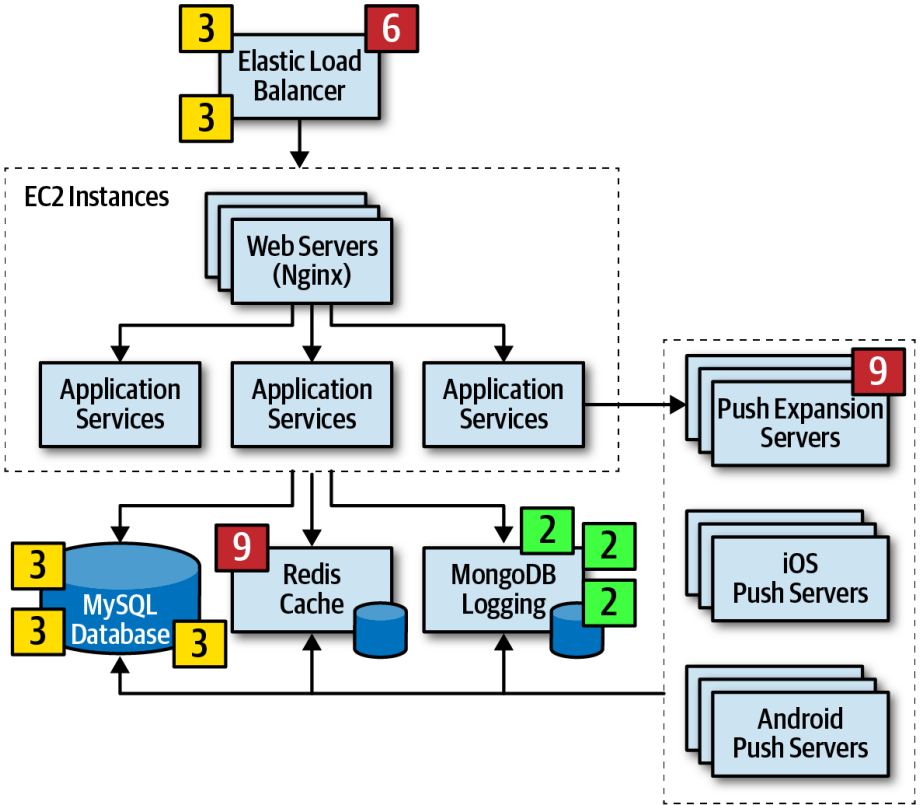
- Quick and fun technique that provides a collaborative way to identify and visualize risks.

Quantify risk



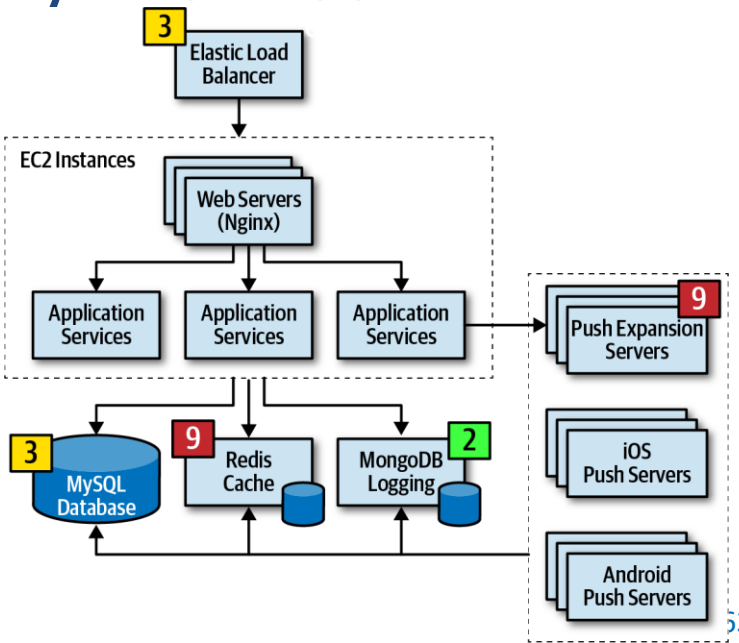
Risk storming

1. Draw the architectural diagrams
2. Team members individually identify risks of the architecture by placing post-it's on the diagrams



Risk storming

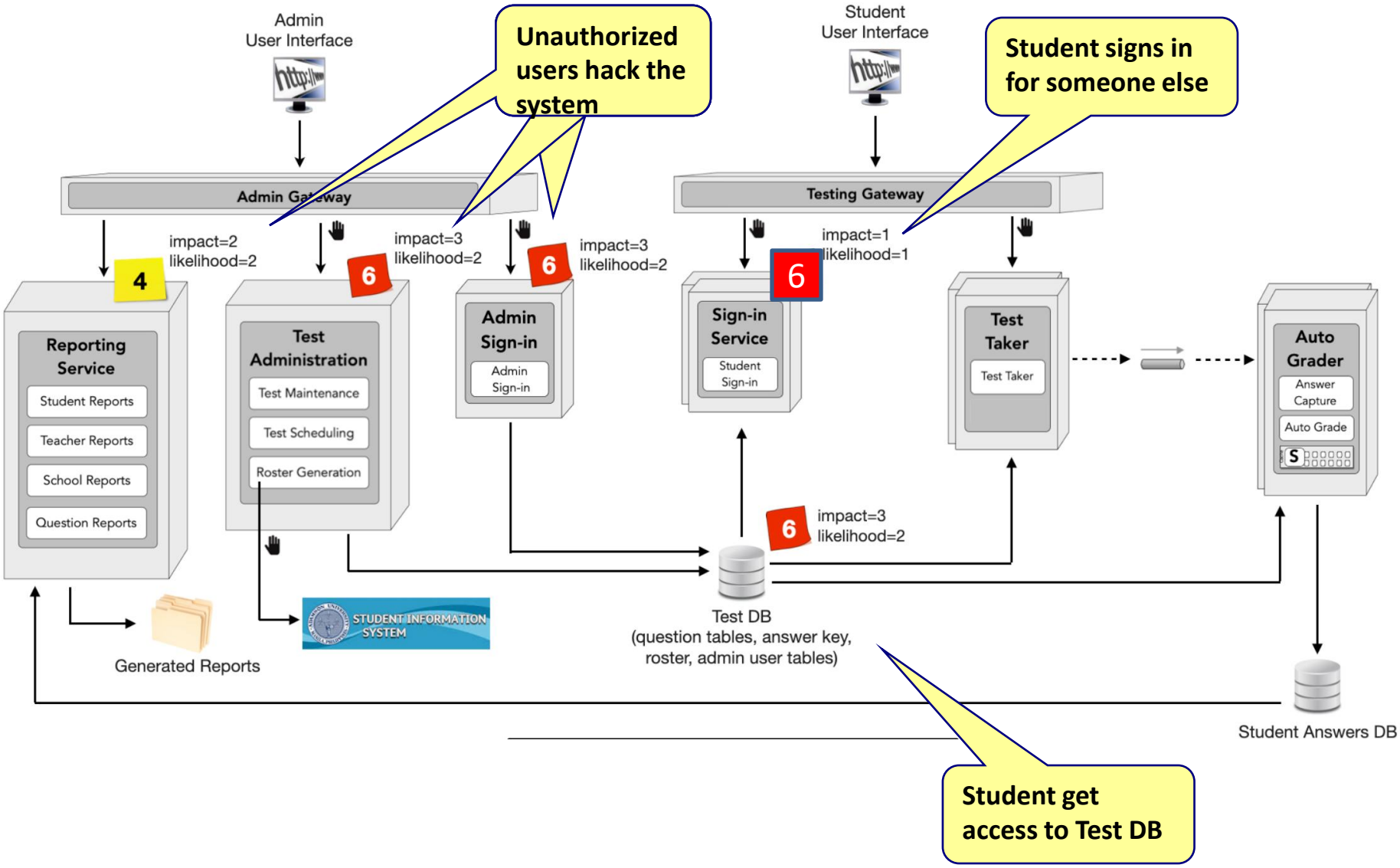
1. Draw the architectural diagrams
2. Team members individually identify risks of the architecture by placing post-it's on the diagrams
3. Discuss result and identify the most important risks



Risk storming

1. Draw the architectural diagrams
2. Team members individually identify risks of the architecture by placing post-it's on the diagrams
3. Discuss result and identify the most important risks
4. Mitigate the risks

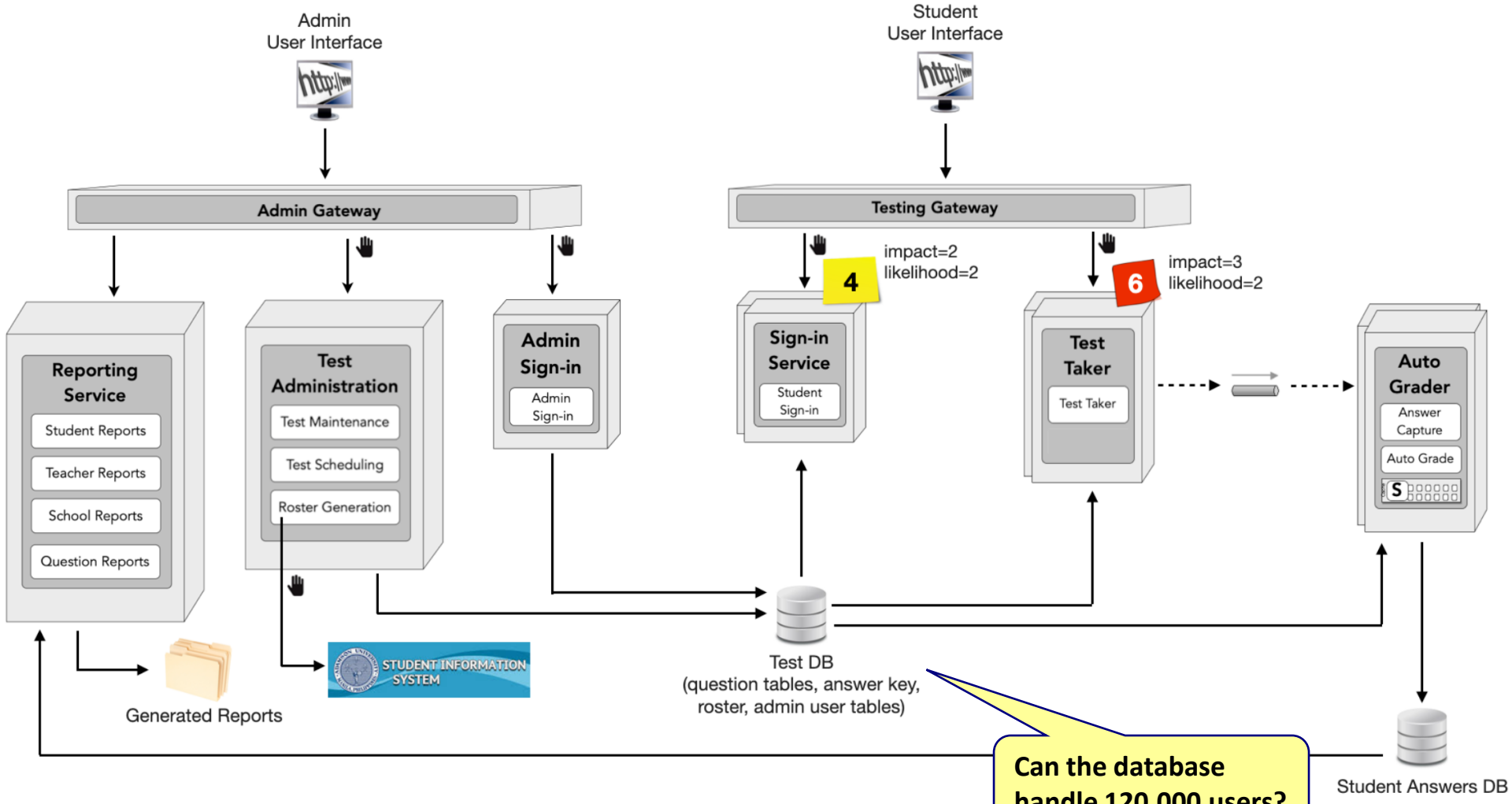
Security risk



Security risk

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

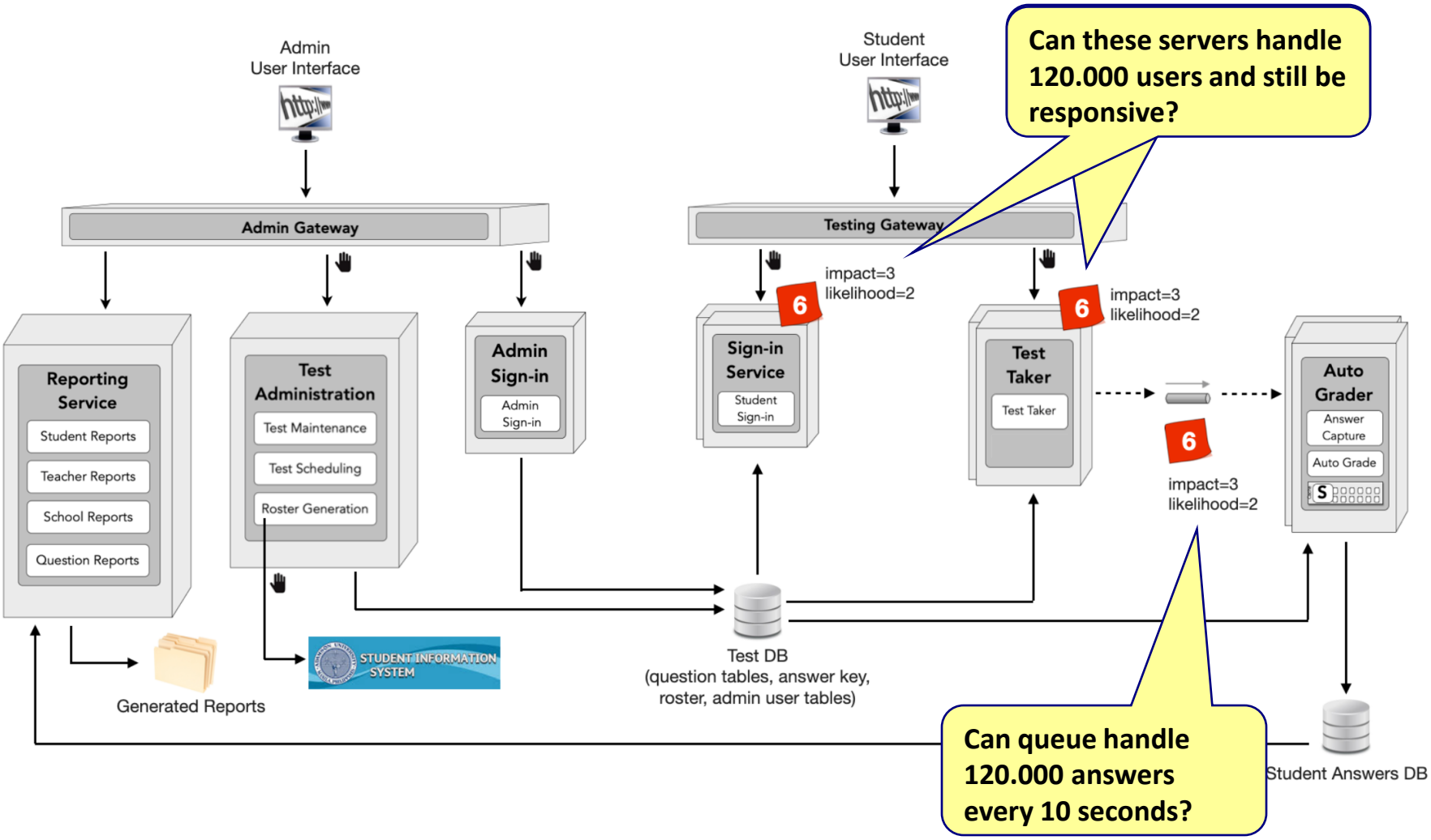
Responsiveness risk



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			

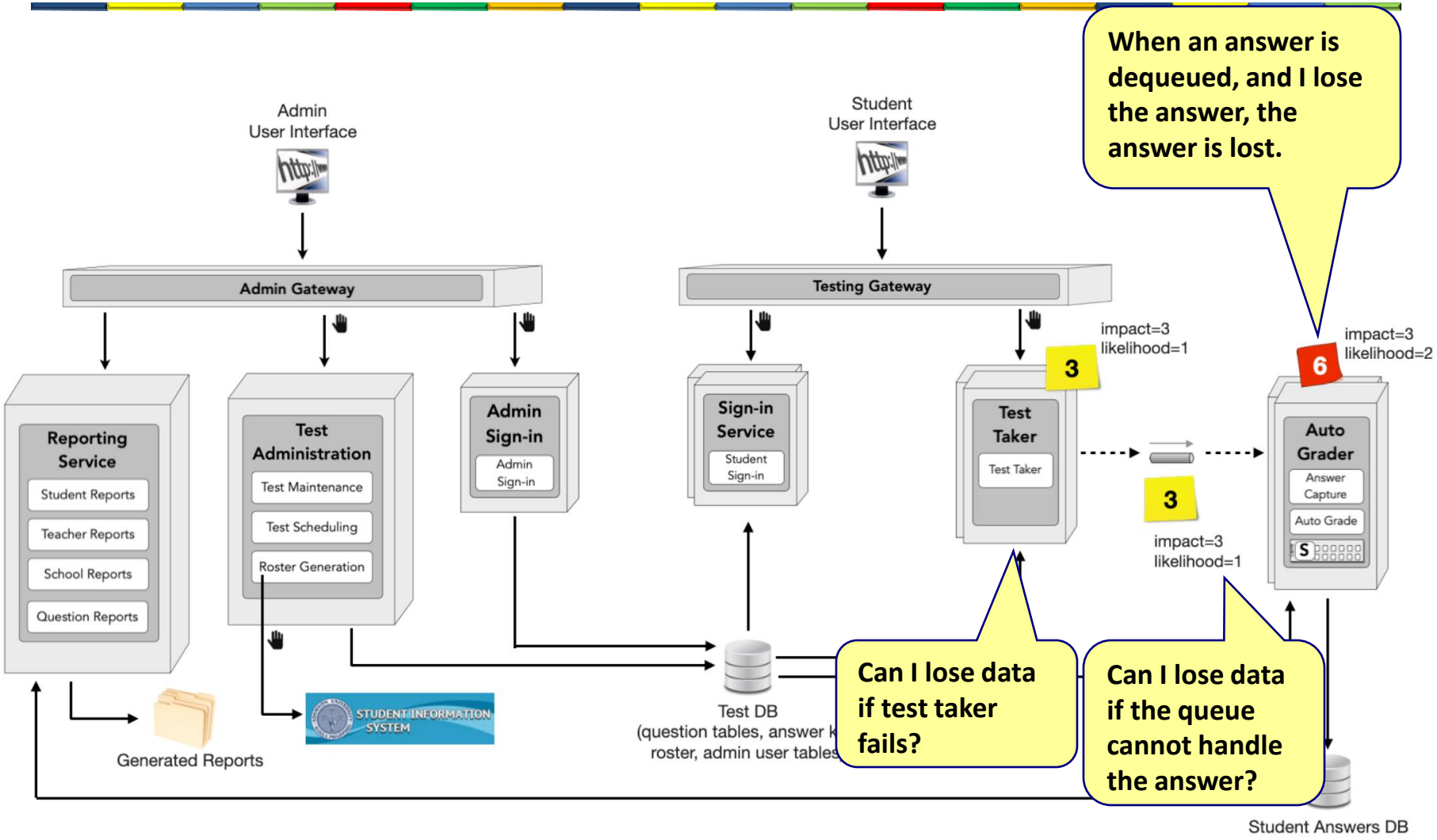
Elasticity risk



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

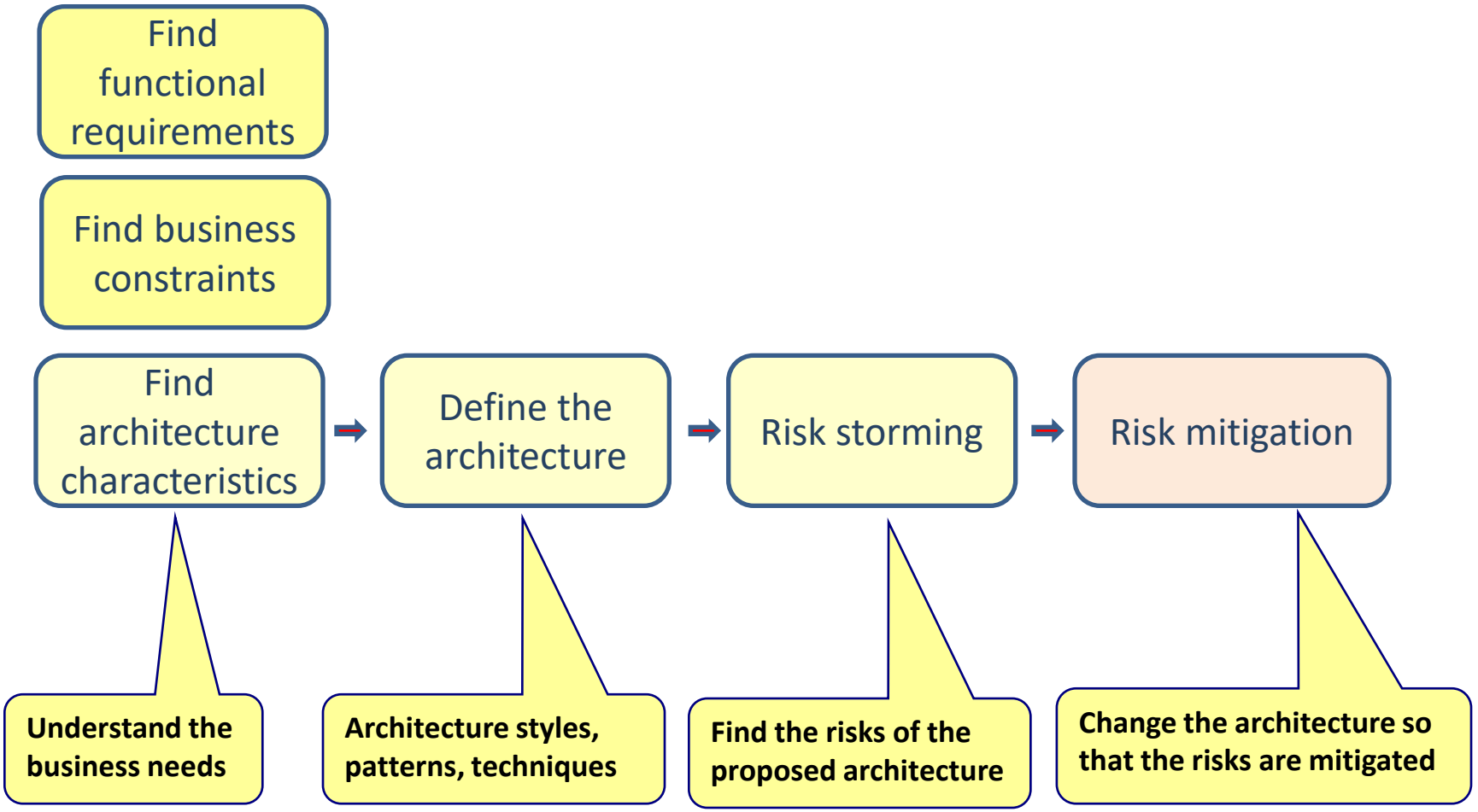


Risk storming result

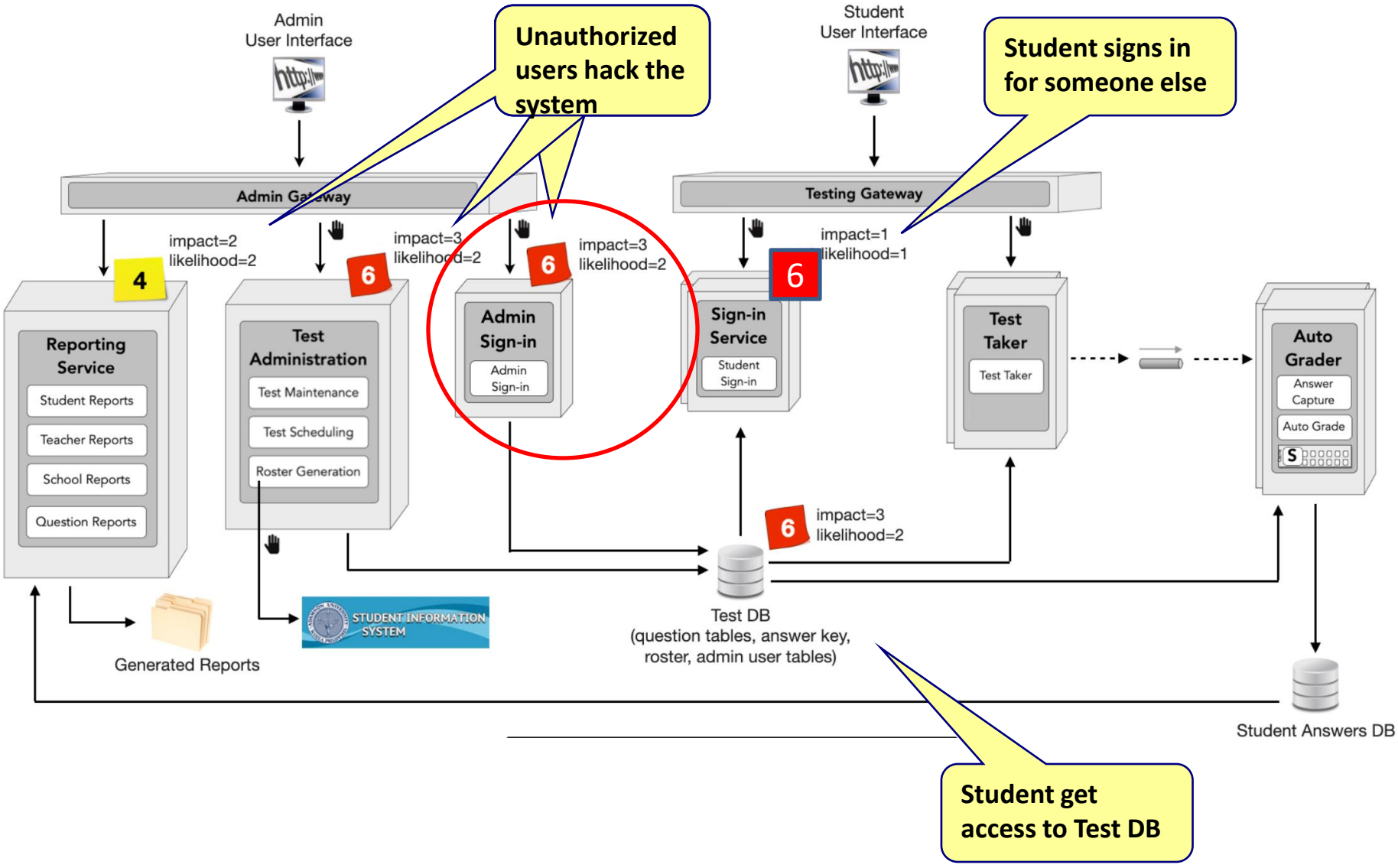
Architecture Risk Assessment	Test Administration	Test Taking	Grading
Data Integrity	1	3	6
Responsiveness	1	6	1
Elasticity	1	6	1
Security	6	6	1
Availability			
Feasibility (cost/time)			
Reliability			

MITIGATING RISK

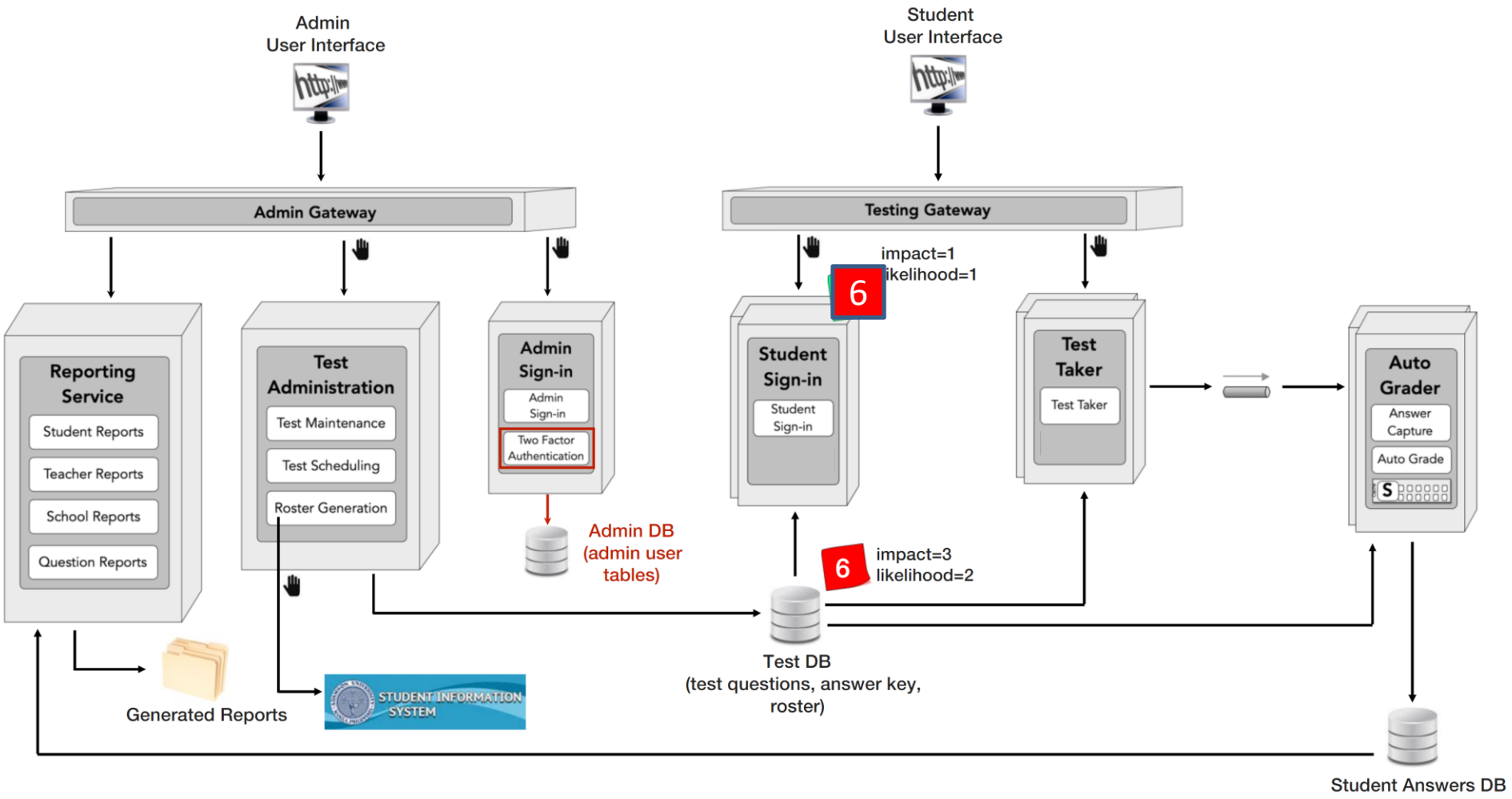
Risk mitigation



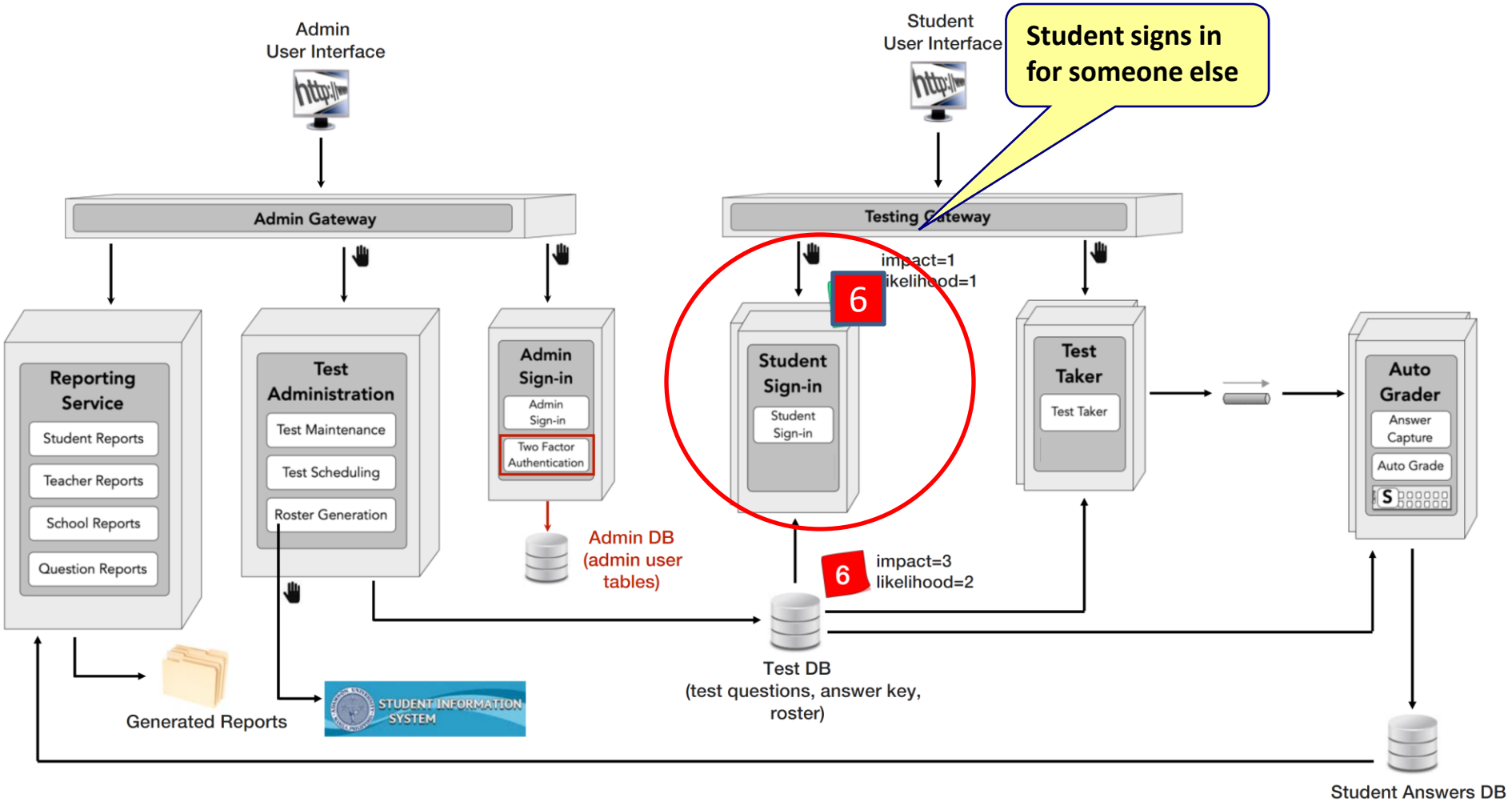
Security risk



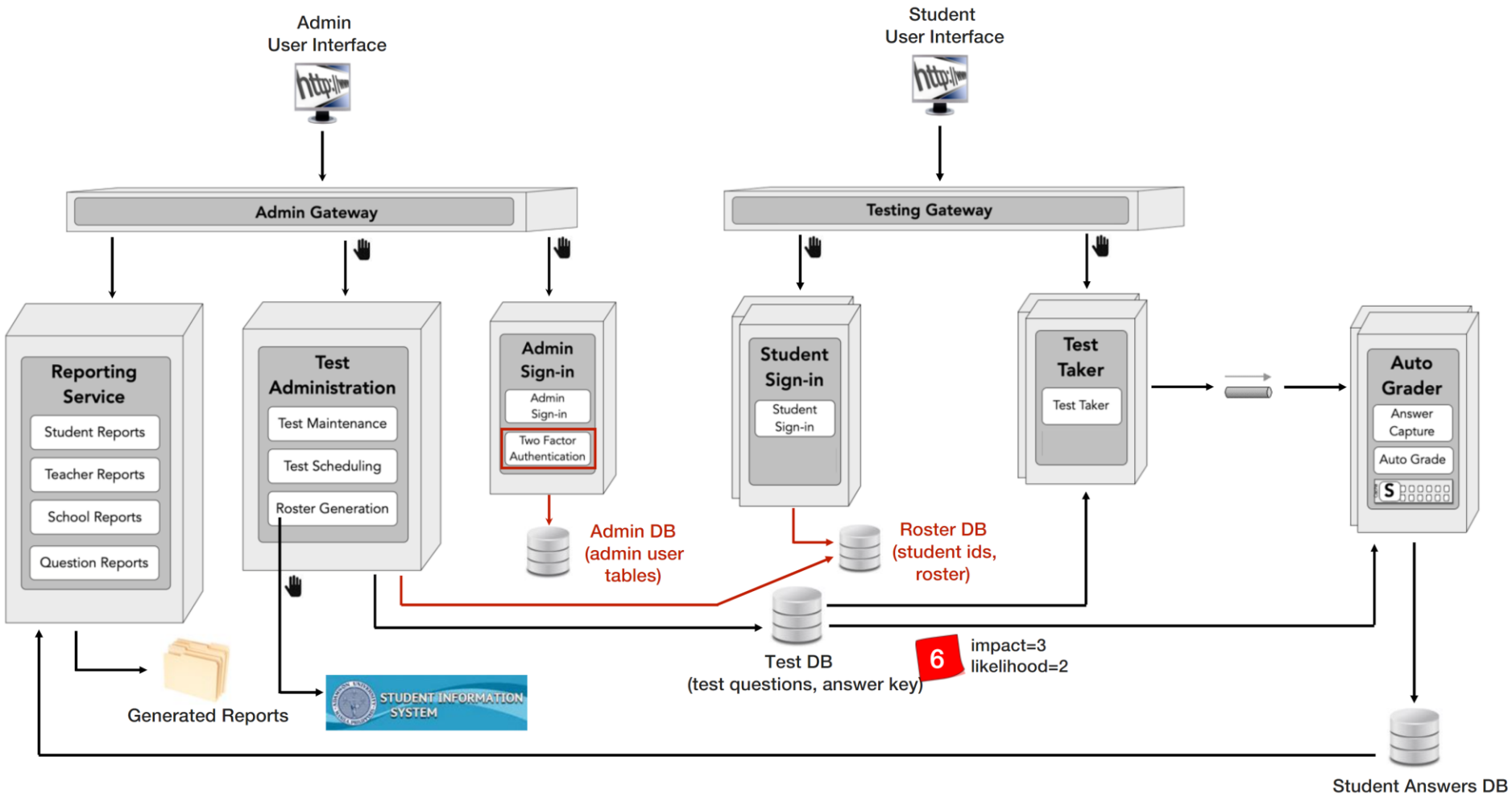
Security risk mitigation



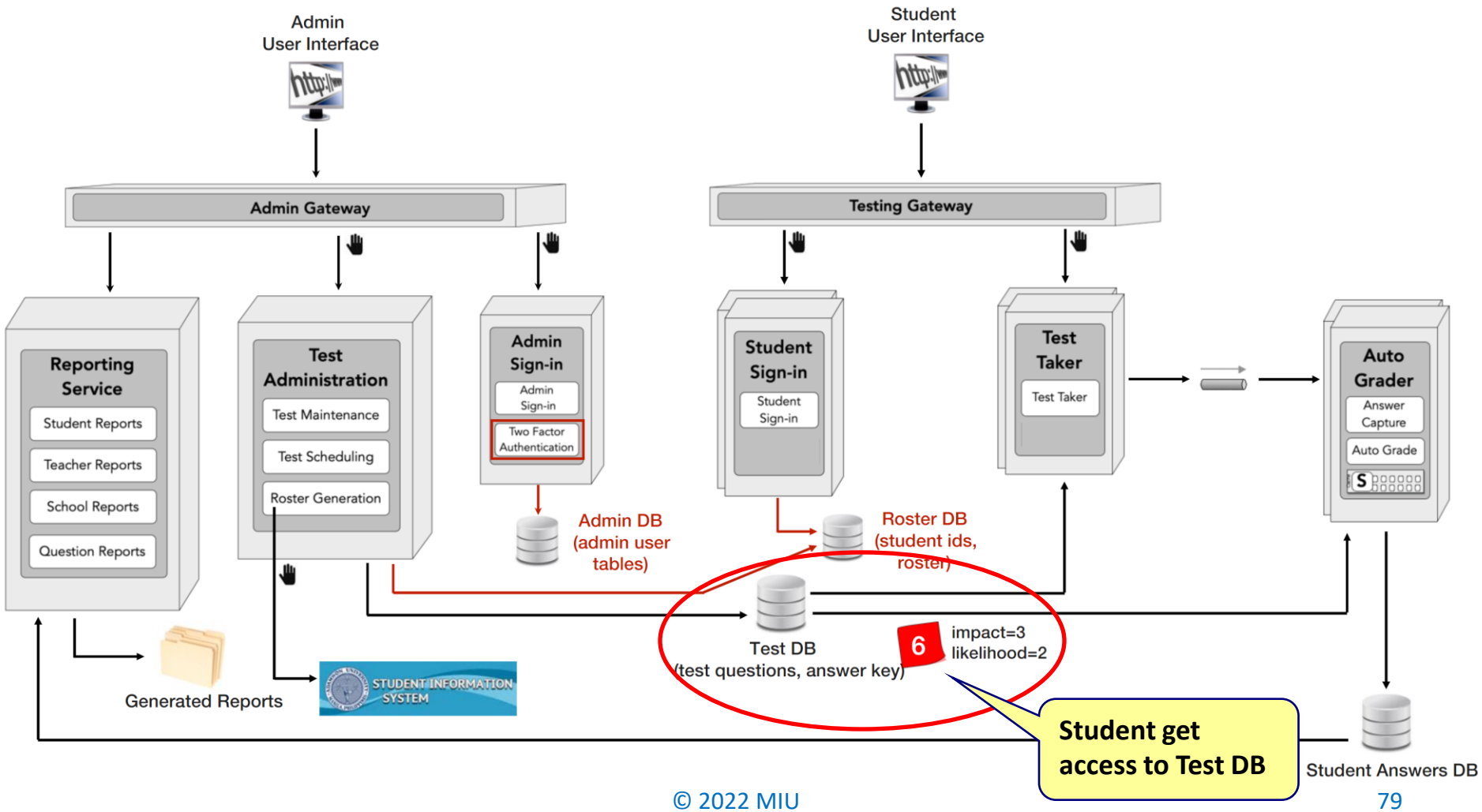
Security risk mitigation



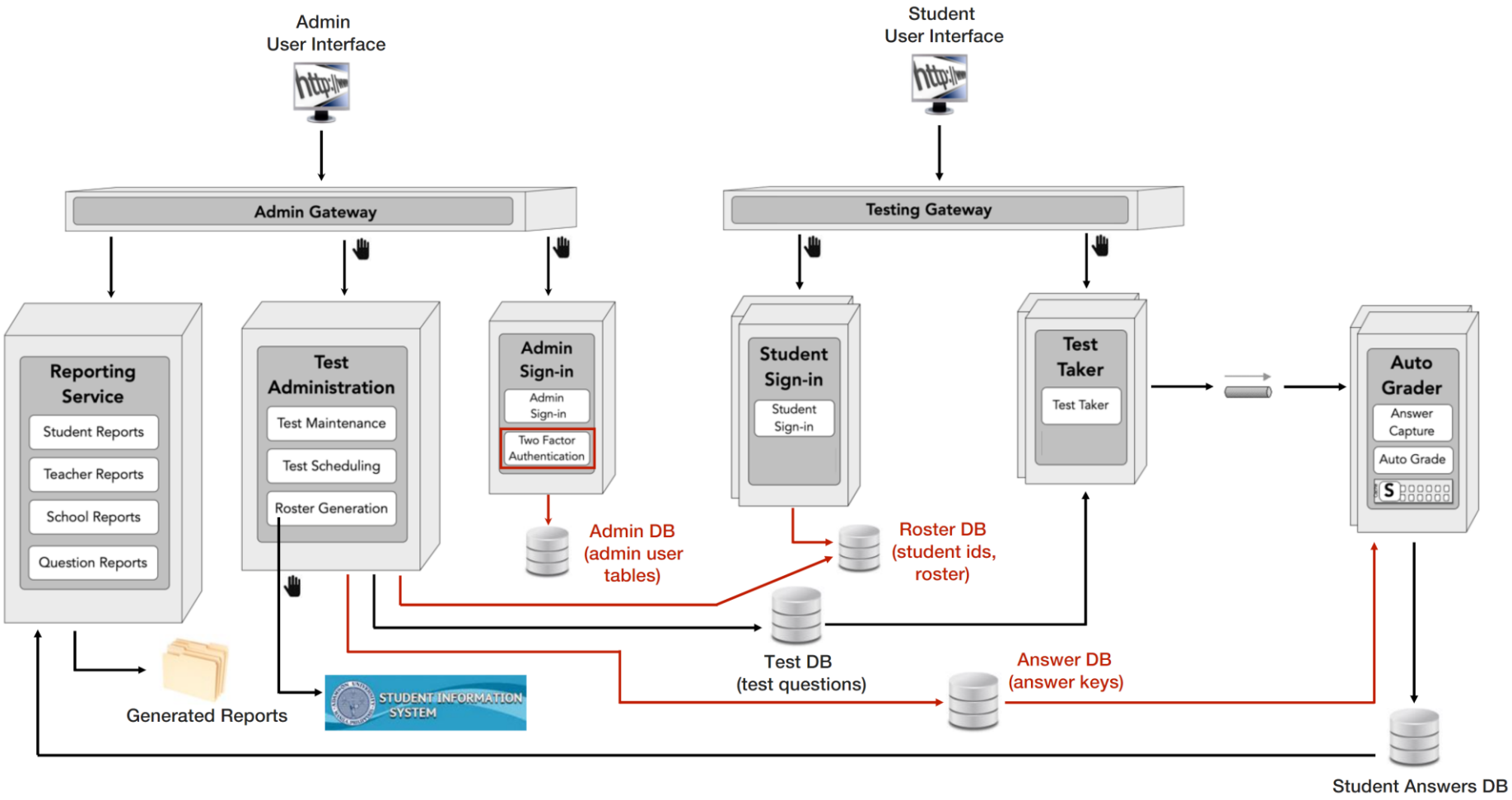
Security risk mitigation



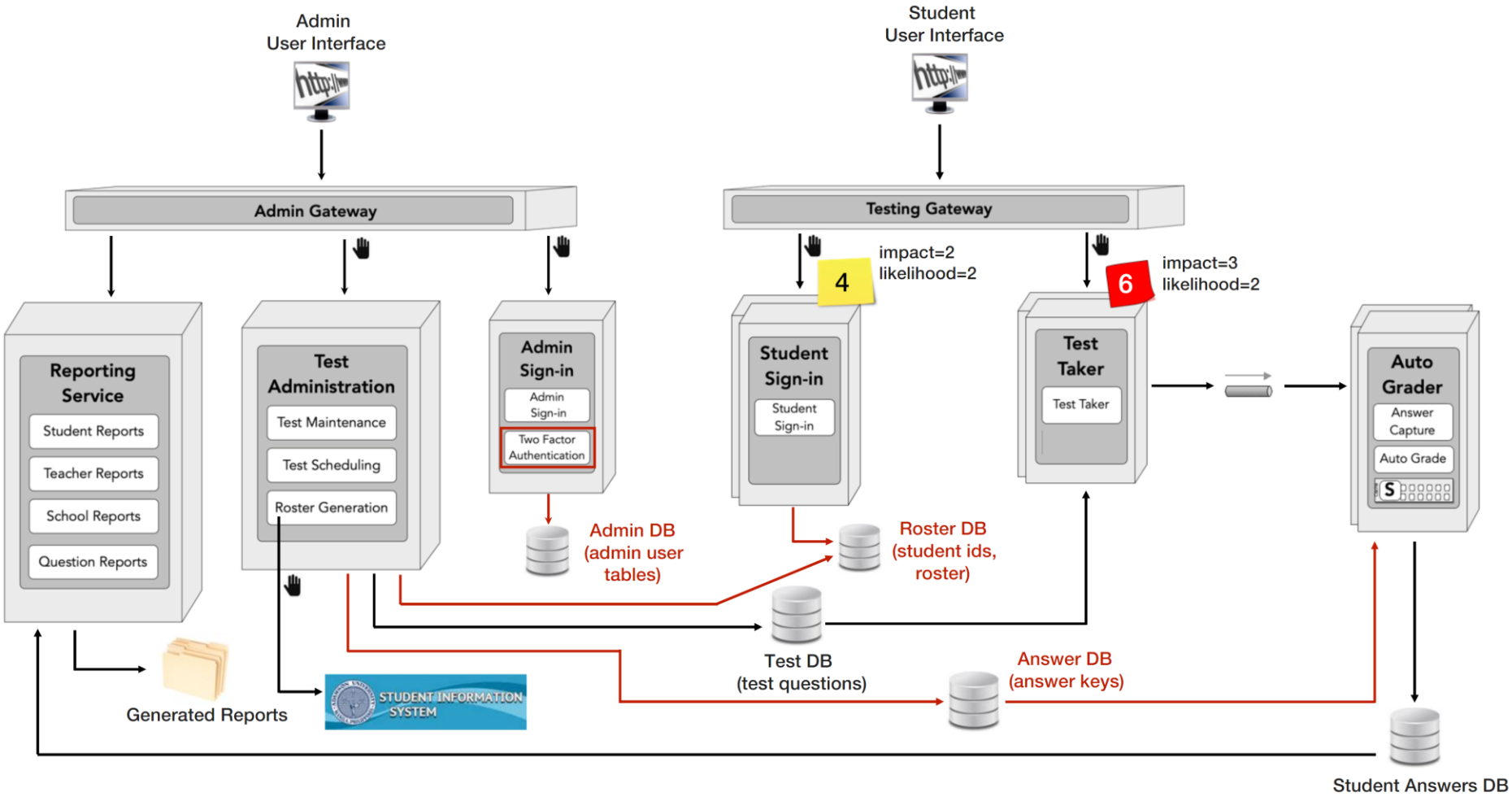
Security risk mitigation



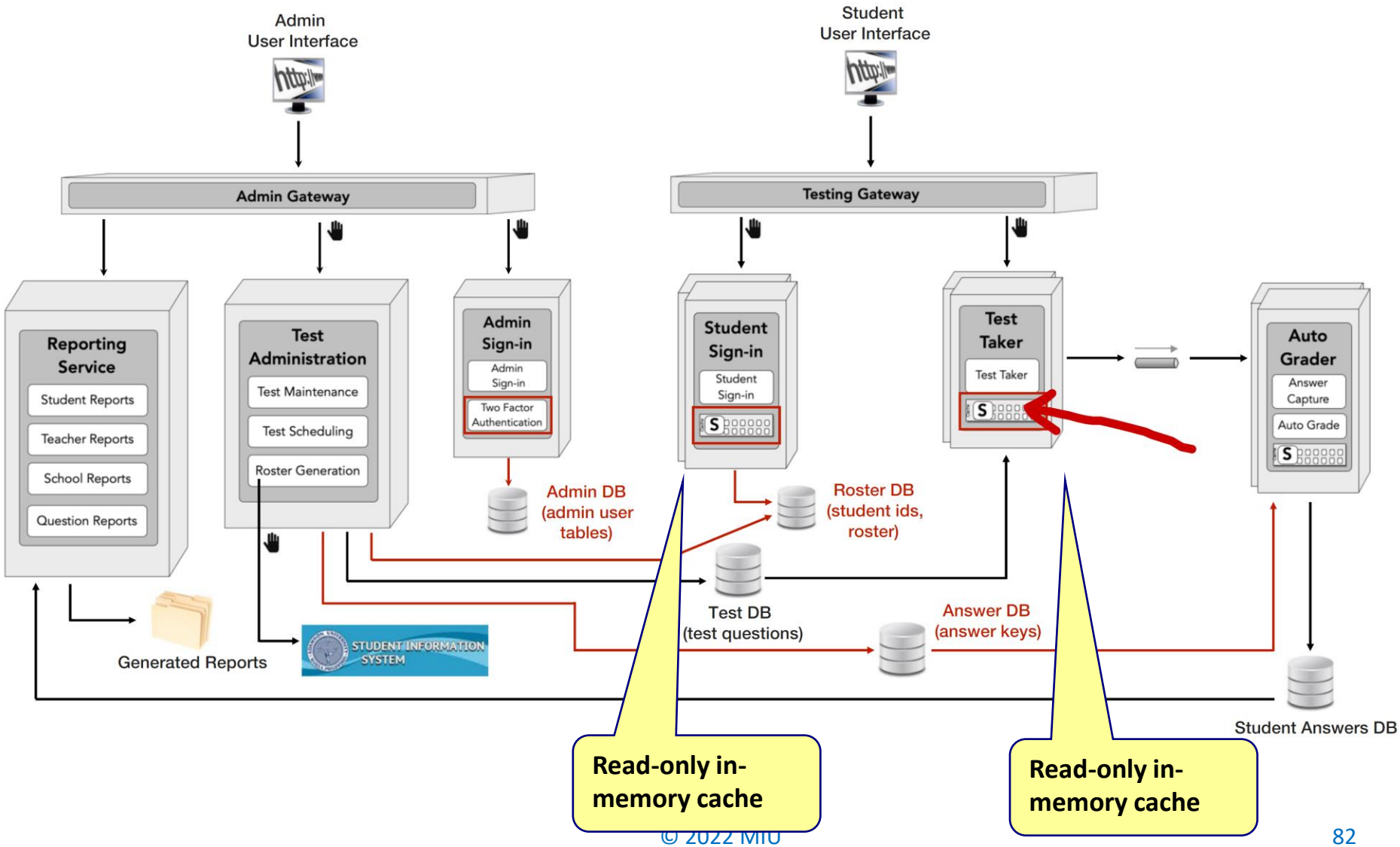
Security risk mitigation



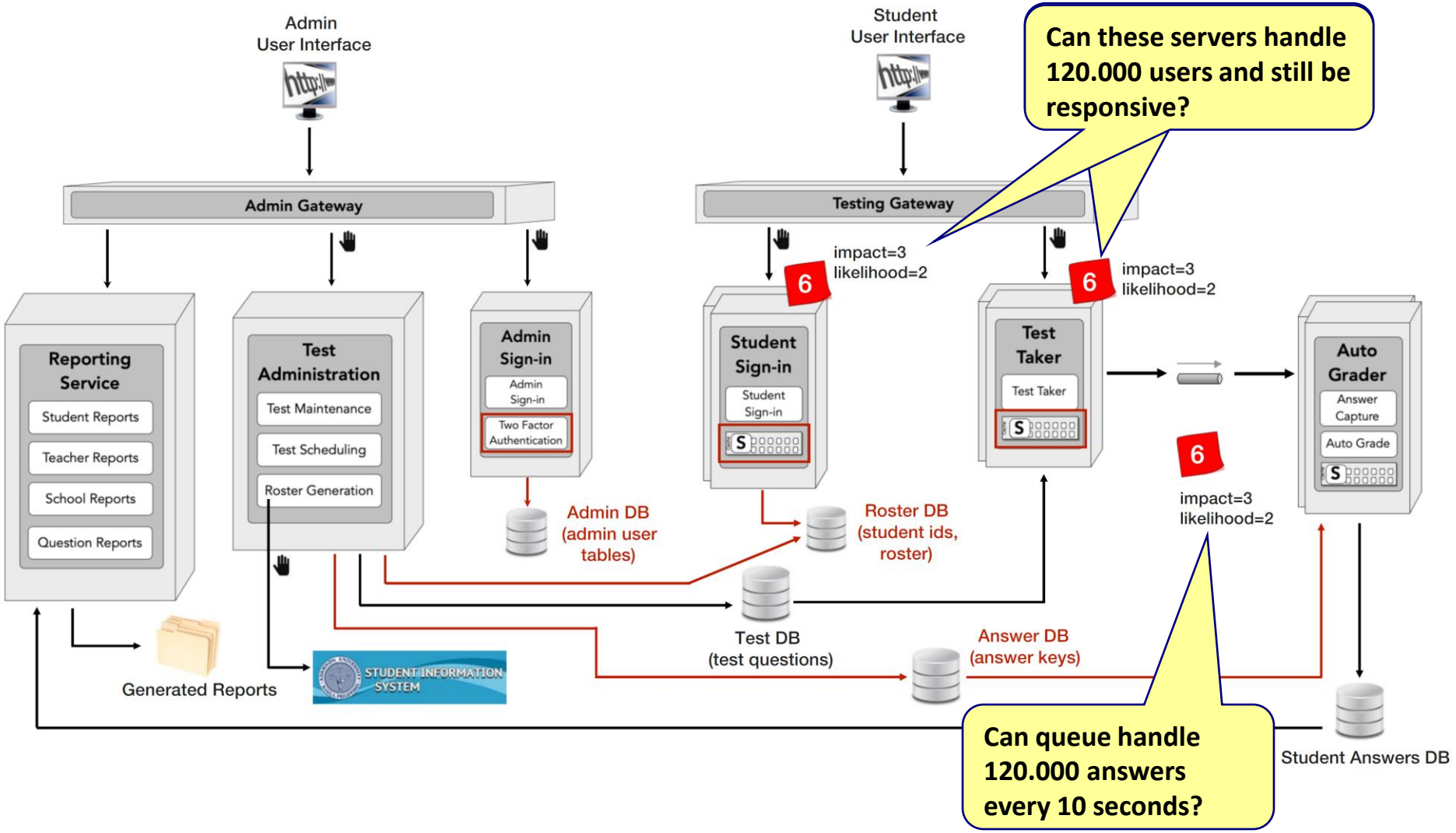
Responsiveness



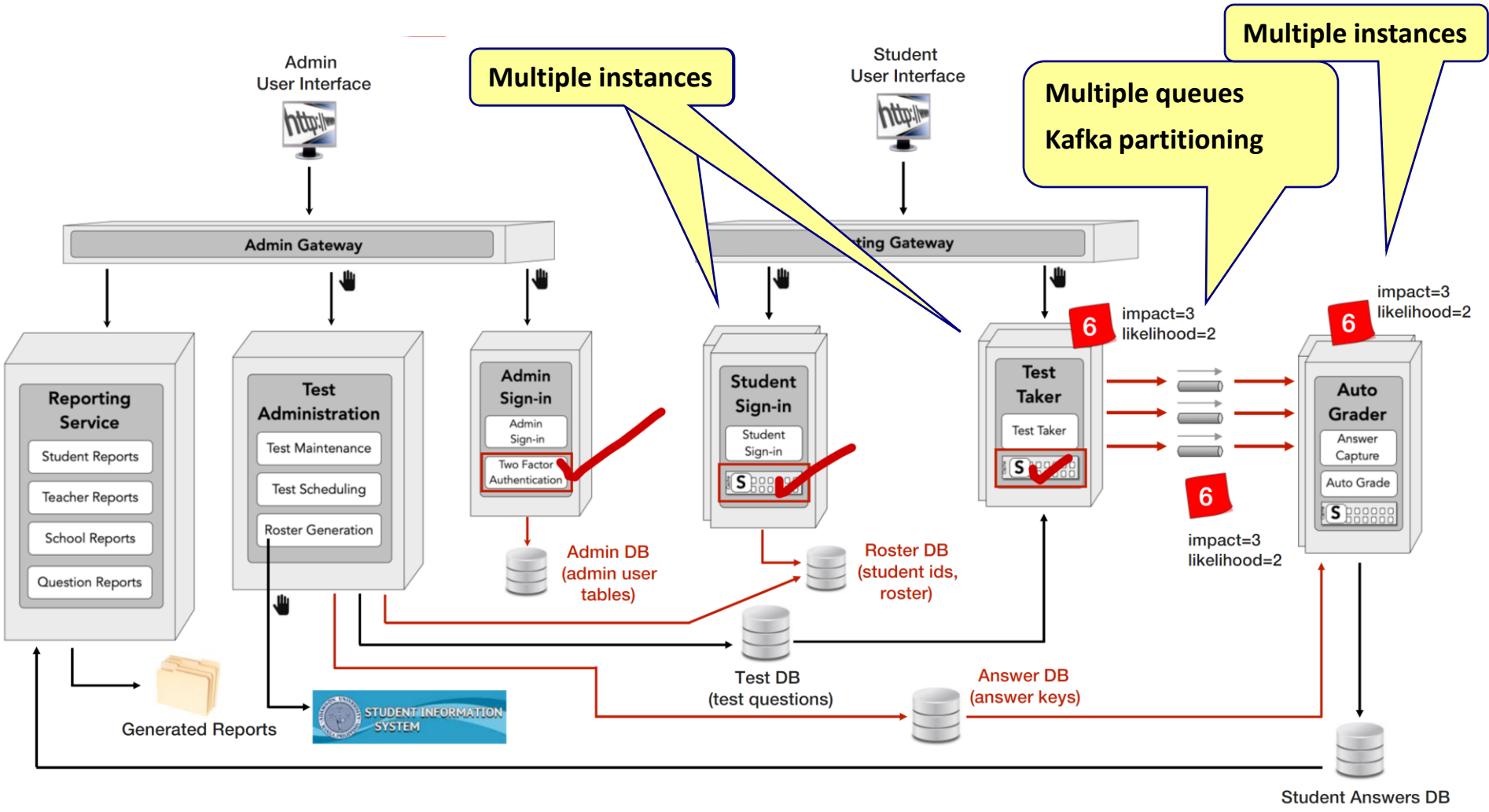
Responsiveness risk mitigation



Elasticity



Elasticity risk migration



Software architecture

- You do this every time the architecture requirements and/or the architecture changes.