8.3.5 Native Airport application to SQL database:

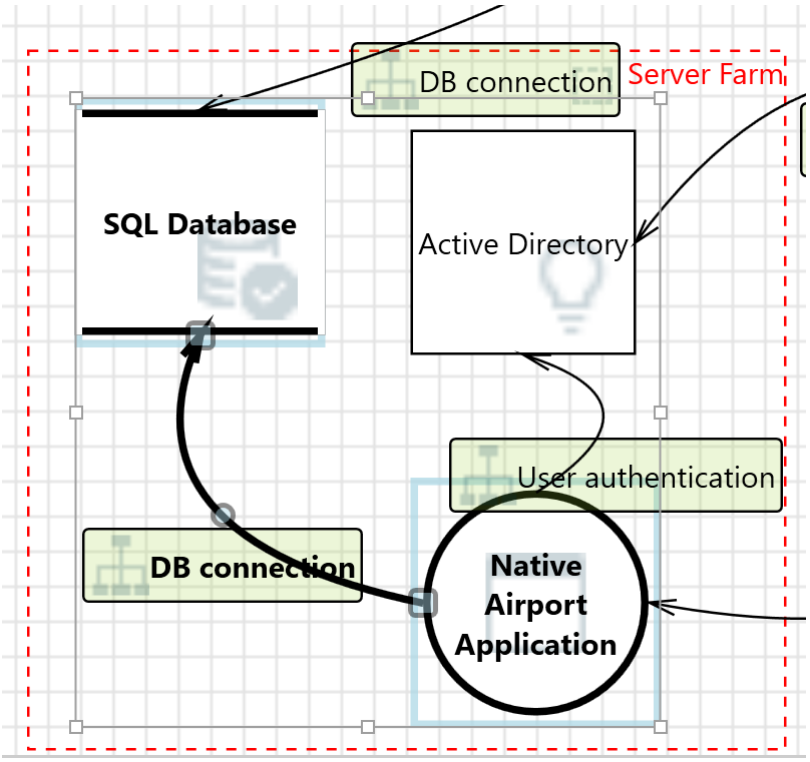

Figure 10: Native application to SQL Database

In Figure 10, the Native airport application to SQL database is depicted. A native airport application could be any application the terminal services team uses to serve passengers, like Flight Information Display Systems (FIDS), Real-time Apron Management Systems, Airport Billing Systems, etc. Listed below are potential threats and corresponding measures to mitigate them.

Potential Excessive Resource Consumption for Native Airport Application or SQL Database

Category:	Denial Of Service
Description:	Does Native Airport Application or SQL Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times when it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock and that they do timeout.
Mitigation:	Implementing rate-limiting mechanisms within the application code will prevent excessive requests from consuming resources. Additionally,

	validating and sanitising all incoming requests to prevent attacks that exploit vulnerabilities, such as buffer overflows or injection attacks.

Potential SQL Injection Vulnerability for SQL Database

Category:	Tampering
Description:	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterised data can be manipulated by a skilled and determined attacker.
Mitigation:	Access to the Native application will be protected with Web Application Firewall. A WAF can inspect the input data from user requests and apply strict validation rules. It checks for suspicious or malicious content, such as JavaScript code or HTML tags, and blocks or sanitises them to prevent SQL injection attacks.

Spoofing of Destination Data Store SQL Database

Category:	Spoofing
Description:	SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.
Mitigation:	User authentication will be done through Single Sign On (SSO) with MFA enabled for all external connections (Grassi et al., 2017).