🔧 ClausePilot – Low-Level Design (LLD)

This document outlines the internal class-level design, data contracts, and service interaction flow for the core ClausePilot services. This level of detail reflects how services are structured and operate at the code level.

🧠 Design Principles & Patterns

ClausePilot follows clean separation of responsibilities with standard design patterns for flexibility and maintainability.

Pattern	Where It's Used	Purpose
Strategy Pattern	`ParserEngine` interface	Easily swap between Tika or Document AI
DTO (Data Transfer Object)	`ContractClauseData`, `UserProfileData`	Structured transport across services
Service Locator	Spring Bean map (`Map<String, ParserEngine>`)	Dynamically resolve parser engine at runtime
Factory (future)	Clause extraction logic	Modular clause matchers per clause type

🧱 Core Class Design

🔹 1. `ParserEngine` (Interface)

Defines the contract for any parsing engine.

```java
public interface ParserEngine {
  String extractText(byte[] fileBytes) throws Exception;
}
TikaParserEngine (Implementation)
@Component("tikaParserEngine")
public class TikaParserEngine implements ParserEngine {
  public String extractText(byte[] fileBytes) {
    return tika.parseToString(new ByteArrayInputStream(fileBytes));
```

3. DocumentAIParserEngine (Implementation)

```
 @Component("documentAIParserEngine")

public class DocumentAIParserEngine implements ParserEngine {

  public String extractText(byte[] fileBytes) throws Exception {

    // Auth and call Document AI client

  }

}
```

4. ClauseExtractorService

```
@Service

public class ClauseExtractorService {

  public ContractClauseData extractClauses(String rawText) {

    ContractClauseData data = new ContractClauseData();

    // Run regex to fill data fields

    return data;

  }

}
```

5. ContractClauseData (DTO)

```
public class ContractClauseData {

  private BigDecimal premiumAmount;

  private Boolean coverageLost;

  private Integer gracePeriodDays;

  private LocalDate policyStartDate;

  private LocalDate expiryDate;

  private String surrenderConditions;

  // Getters, Setters, Constructors

}
```