**Services:**

**1. Old Services (AngularJS):**

- In AngularJS (Angular 1.x), services were typically created using the service or factory methods. These services were registered with Angular's dependency injection system and could be injected into controllers, directives, and other services.

- Services in AngularJS were often used for tasks like data fetching, business logic, or sharing common functionality across multiple components.

- However, services in AngularJS lacked some of the features and optimizations found in Angular, such as tree-shakable providers and Ahead-of-Time (AOT) compilation.

**2. New Services (Angular 2+):**

- In Angular 2 and later versions, services are still used for the same purposes but are implemented differently and offer more features and optimizations.

- Services in Angular are typically created using TypeScript classes decorated with the @Injectable() decorator. This decorator marks the class as a candidate for Angular's dependency injection system.

- Angular services can be provided at various levels of the application's injector hierarchy, allowing for flexibility in terms of scoping and sharing of services.

- With Angular's modular architecture, services can be organized within feature modules and lazy-loaded modules, improving the overall structure and maintainability of large applications.

- Angular's Ahead-of-Time (AOT) compilation and tree-shakable providers enable better performance and smaller bundle sizes by eliminating unused services and dependencies from the final production code.

- Additionally, Angular provides built-in support for Observables and RxJS, making it easier to handle asynchronous operations and data streams within services.

Overall, while the core concept of services remains the same between AngularJS and Angular, the implementation and capabilities have evolved significantly in newer versions of the framework, offering developers more powerful tools for building scalable and maintainable applications.

**Components:**

1. AngularJS (1.x): The first version, based on MVC architecture, uses JavaScript. It has two-way data binding and focuses on building dynamic web apps.

2. Angular (2+): Completely rewritten from scratch in TypeScript. It's component-based and uses a hierarchical dependency injection system. Introduced features like Angular Universal for server-side rendering, Ahead-of-Time (AOT) compilation, and improved performance.

3. AngularJS (1.x) vs. Angular (2+): AngularJS uses controllers and $scope, while Angular (2+) uses components and directives. AngularJS uses digest cycle for change detection, while Angular (2+) uses a more efficient mechanism called Zone.js.

4. Angular 2, 4, 5, 6, 7, 8, 9, 10, 11, 12: These versions brought incremental improvements, features, and optimizations. Angular now follows Semantic Versioning, meaning major updates bring breaking changes.

5. AngularJS vs. Angular (2+): AngularJS is less performant, less modular, and doesn't support TypeScript. Angular (2+) is more performant, modular, supports TypeScript, and has a better tooling ecosystem.