The technical differences between ZeroDev's Kernal plugins, SAFE Modules, and EIP-6900?

First, let us understand what the plugins are and why we need them in a dApp.
There are three main reasons for plugins to exist.

Why do we need plugins?

Security: Since smart contracts are insecure, a plugin can be used to improve security by implementing timelocks and access control. A plugin can stop malicious unauthorized transactions and require a transaction to be approved by multiple parties before execution.
Flexibility: Plugins can add new features to smart contracts and extend their functionality. It's hard for non-tech folks to interact with smart contracts directly, so Plugins can be used to implement a voting system that allows users to vote on proposals without interacting with smart contracts directly by providing a user-friendly interface.
Interoperability: Plugins can be used to interact with different smart contracts or chains. For this, plugins have to define standard interfaces that allow plugins to interact with each other. By defining a standard interface, a plugin implementing the voting system can interact with a plugin implementing the user interface.

ZeroDev's kernel plugins

ZeroDev's Kernel plugins are Solidity contracts used to extend the functionality of Kernel-based accounts. Kernel-based accounts are a new type of account being developed on the Ethereum blockchain. Kernel-based accounts are made of three components. Validation, execution, and hooks. As the name suggests, first two are for validating and executing transactions. While hooks are used for extending the functionality of an account. They can be used with any dApp or wallet that supports solidity contracts making them highly flexible and can add new features such as timelocks, access control, and hooks. They are implemented as separate plugins and are highly interoperable with any kernel-based account. One downside is they don't have many security features built in.

SAFE modules

SAFE modules are a set of Solidity contracts that provide many security features for smart contracts for dApps or wallets compatible with Solidity contracts. They are implemented as separate modules, which makes them easier to use and implement. SAFE Modules are less flexible than ZeroDev's Kernel plugins because they are designed to provide a specific set of security features and cannot be easily modified. However, they are highly interoperable and used in any smart contract that supports the SAFE standard. As you might have guessed, they are highly secure with multiple security features built in, such as signature verification and access control.

EIP-6900

EIP-6900 is a proposal for a standard for modular smart contract accounts and account plugins. This standard would allow composable logic within smart contract accounts and make plugins easier to develop securely. EIP-6900 is not a specific implementation but rather a set of guidelines that developers can follow. It is designed to be flexible. It allows for a wide variety of plugins to be implemented and does not impose any specific security requirements. However, it provides a framework for implementing security features, assuming that most plugins will implement some level of security. EIP-6900 is designed to be interoperable. It defines a standard interface for plugins, making it easy to develop plugins compatible with different accounts.

Conclusion

ZeroDev's kernel plugins are highly flexible and interoperable but not much secure. In contrast, SAFE modules are highly secure but not so flexible. EIP-6900 is a compromise between the two. It is still under development, but it has huge potential and likely to be the most widely adopted approach.