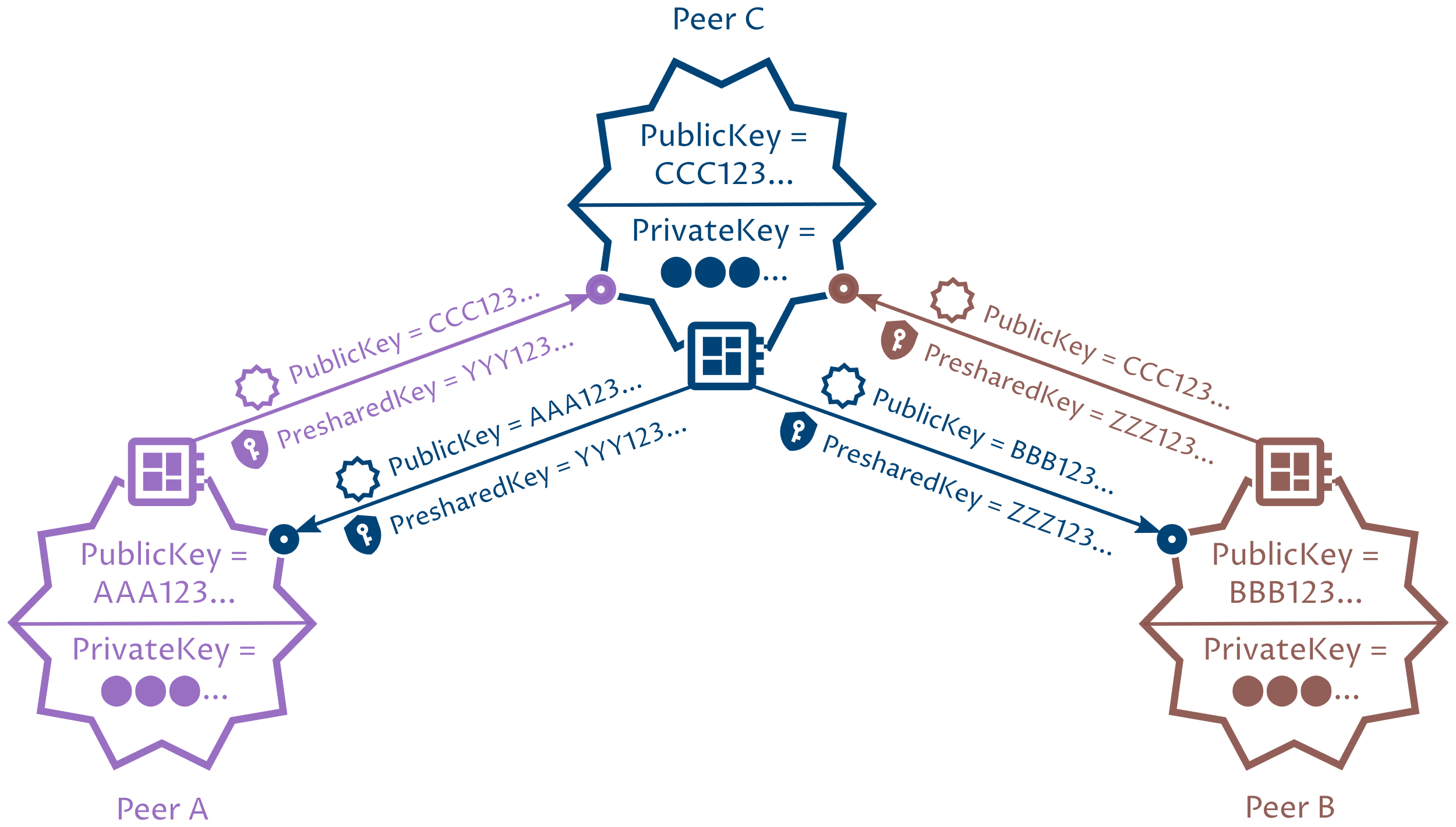
**РАЗРАБОТКА VPN-СЕРВЕРА**

Для начала импортируем в проект класс провайдера настроек[8], повышающий абстракцию использования файла appsettings.json и прочих, позволяя использовать его как набор C#-объектов в классах остальных сервисов. Этот класс был разработан значительно ранее.

В начале разработки был допущен оверинжининг – с попыткой добавить в приложение базу данных и 3 различных типа аккаунтов для взаимодействия с API, что не соответствует изначальной концепции легковесности docker-сборки VPN-сервера. Изначальный код сохранен в истории репозитория и удален. Вывод состоит в том, что не стоит делать умными все части сложной системы, а лучше сделать таковой одну часть, которая будет взаимодействовать с остальными по концепции master-slave.

На этапе прототипирования возник вопрос использования pre-shared key в пирах Wireguard. Данные ключи обеспечивают симметричное шифрование, в дополнение к асимметричному (по открытому ключу). По большому счету, данный функционал предназначен для защиты от атак квантовых компьютеров[9] и не будет добавлен в решение:

*« Importantly, you don’t* need *to use preshared keys with WireGuard — there are no known practical attacks against the WireGuard cryptosystem, even without preshared keys. However, because WireGuard key pairs use elliptic-curve cryptography (ECC), in the future it might be possible for quantum computers to “crack” those key pairs (where the quantum computer would be able to calculate the private key from the public key). This would enable an organization with a quantum computer to decrypt any encrypted WireGuard traffic that it had previously captured and stored.*

*Using preshared keys means that even if an attacker calculated (or was able to steal) the private keys used by a WireGuard connection, she still wouldn’t be able to decrypt the connection’s encrypted WireGuard traffic. To do so, she would also have to steal the preshared key (a randomly-generated preshared key would not be “crackable” by quantum computers, nor by any other means). »*

Опишем теперь работу прототипа ASP-приложения. В приложении есть всего несколько моделей данных, одно мидлваре, один сервис, один контроллер. Поскольку модели данных все