Antes de mais nada, deixo a ressalva que a aplicação aqui desenvolvida, foi feita para ser minimamente útil no menor intervalo de tempo possível com o custo de não ser seguido as melhores práticas do mercado para uma aplicação de alta qualidade.

Dito isto, discutirei agora possíveis melhorias a serem feitas para garantir a maio qualidade nos pontos de resiliência, segurança e simultaneidade da aplicação, assim como uma maior performance das operações do banco de dados.

Para aumentar a resiliência da aplicação, maiores tratativas de erros devem ser implementadas, como uma melhor validação do corpo dos requests, além da criação de testes unitários para as ViewSets(atualmente foram criados alguns testes para fins de demonstração da capacidade).

Outro ponto a ser melhorado é a implementação de métodos assíncronos para atender ao requisito de simultaneidade, dessa forma ações diferentes em cada ViewSet podem ser feitas simultaneamente.

Para garantir a segurança das APIs, melhores políticas de token devem ser criadas, como limite de validade do token, e até mesmo validações extras podem ser criadas, como por exemplo a verificação se o IP de solicitação do Token é o mesmo sendo utilizado no momento, e limite de chamadas por segundo que cada usuário pode fazer, limitando assim a possibilidade de ataques DDOS. Prevenção à SQL Injection já é uma ferramenta presente no ORM do Django.

Um último ponto seria a performance com relação ao banco de dados, e é exatamente nesse campo que mais ações podem ser feitas, inclusive a implementação da modelagem proposta no .dbml (a modelagem do banco atual não segue a do .dbml por questões de facilidade de implementação). A criação de index nas tabelas pode ser usada para melhorar a performance, tabelas como a tabela de Keywords e Comentários podem ser especialmente beneficiadas com essa criação, já que são tabelas com potencial de ficarem massivas e com queries complexas em caso de ativação da possibilidade de exclusão e edição de conteúdos em geral. Por falar em queries complexas, ao invés de utilizar essas queries, o ideal seria utilizar Stored Procedures, que atualmente nessa fase da implementação não se fazem necessários.