Masz rację! **Email** i **EmailSettings** nie znajdują się w warstwie domeny, ponieważ **nie są podstawowymi elementami logiki biznesowej** aplikacji. W Clean Architecture rozdzielamy to, co jest centralne dla domeny (logiki biznesowej), od tego, co jest specyficzne dla infrastruktury. Spójrzmy, dlaczego:

**1. Email i EmailSettings nie są częścią logiki biznesowej (domeny):**

* **Domena** to serce aplikacji, które zajmuje się **głównymi regułami i modelami biznesowymi**. Obejmuje encje i logikę, które są kluczowe dla biznesu, takie jak **kategorie**, **wydarzenia**, czy **zamówienia** w Twojej aplikacji.
* **Email** i **EmailSettings** dotyczą komunikacji i konfiguracji technicznej, co oznacza, że są one bardziej związane z **infrastrukturą** aplikacji, a nie z zasadami biznesowymi.
* **Email** to mechanizm, który używasz do wysyłania wiadomości — jest to operacja infrastrukturalna, a nie operacja biznesowa. Twoja aplikacja może działać bez wysyłania e-maili, ale bez obsługi wydarzeń, zamówień czy kategorii (czyli logiki biznesowej) straciłaby sens.

**2. Dlaczego EmailSettings są w warstwie aplikacji lub infrastruktury:**

* **EmailSettings** (jak widzimy na Twoim obrazku) to klasa, która zawiera dane konfiguracyjne dotyczące wysyłania e-maili — takie jak **ApiKey**, **FromAddress**, i **FromName**. To są szczegóły techniczne, które są potrzebne do realizacji wysyłania e-maili, ale nie są związane z samą logiką biznesową Twojej aplikacji.
* Te informacje są **specyficzne dla infrastruktury**, ponieważ mówią, **jak** wysyłać e-maile (np. za pomocą zewnętrznego dostawcy usług e-mail), a nie **dlaczego** te e-maile powinny być wysyłane.

**3. Operacje na e-mailach są obsługiwane w warstwie infrastruktury:**

* **EmailService.cs** prawdopodobnie znajduje się w warstwie infrastruktury, ponieważ to tam implementujesz logikę techniczną dotyczącą tego, jak wysyłać e-maile. **EmailService** korzysta z klasy **EmailSettings**, aby pobrać odpowiednie informacje, jak np. adresy e-mail, ale nie jest to część domeny.

**4. Kiedy umieścić coś w domenie, a kiedy w infrastrukturze?**

* Jeśli coś jest **niezależne od technologii**, ale stanowi **kluczową logikę biznesową** (np. obliczanie rachunków, zarządzanie zamówieniami), to umieszczasz to w **domenie**.
* Jeśli natomiast coś dotyczy **konkretnej technologii**, ale **nie jest krytyczne dla zasad działania biznesu**, umieszczasz to w warstwie **infrastruktury** lub **aplikacji**. W tym przypadku **wysyłanie e-maili** to operacja techniczna, a nie biznesowa.

**Podsumowanie:**

* **Email** i **EmailSettings** znajdują się poza warstwą domeny, ponieważ dotyczą szczegółów technicznych związanych z wysyłaniem wiadomości, a nie z kluczową logiką biznesową.
* Kluczowa logika biznesowa, która znajduje się w domenie, powinna być niezależna od technologii, a wysyłanie e-maili jest stricte zależne od technologii (np. API dostawcy e-maili).
* Dlatego te elementy umieszczasz w warstwie **infrastruktury** lub **aplikacji**, a nie w domenie.

Twoja intuicja tutaj była bardzo trafna!