

## **Key Flexibility Design Principles**

Making the NZ market attractive for private investment

#### 1) Level Playing Field

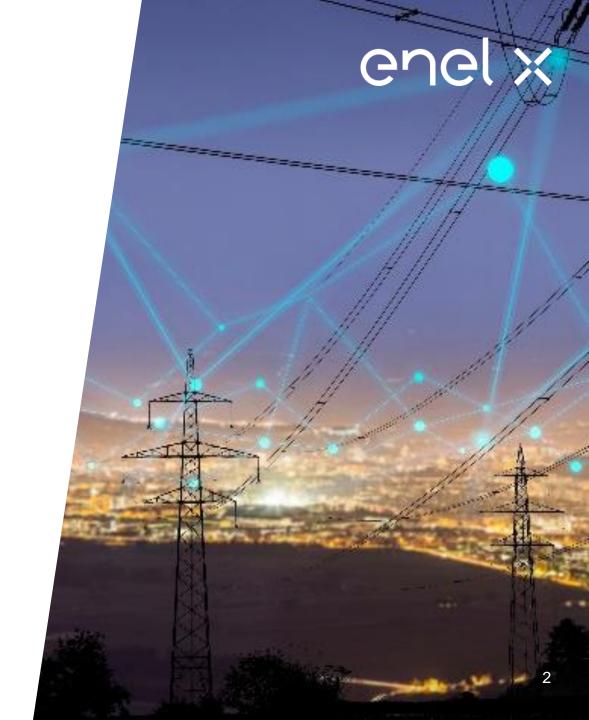
- Confidence in cost allocation
- Transparency

#### 2) Confidence in correct utilisation

- It's being used for its clear purpose
- Transpower-owned DERMS can not influence market outcomes

#### 3) Investment Signals

- Revenue certainty
- Scale
- Value-stacking ability







- Share concerns over Transpower operating FM/S
  - Sceptism of competition when the buyer can contract itself, no matter what ring-fencing rules are in place
- Clear rules around ring-fencing are required from the start. There's concerns about the terminology in the memo ('monitoring progress', 'confirmed', etc.) which we believe doesn't fulfil EA's strategic ambition for low emissions energy

...implementing market rules that give investors confidence...

...stable investment environment with robust rules...

Transpower have said their committed to operate appropriately, but can they be incentivised to?

Further consideration



Discretion breeds nervousness:

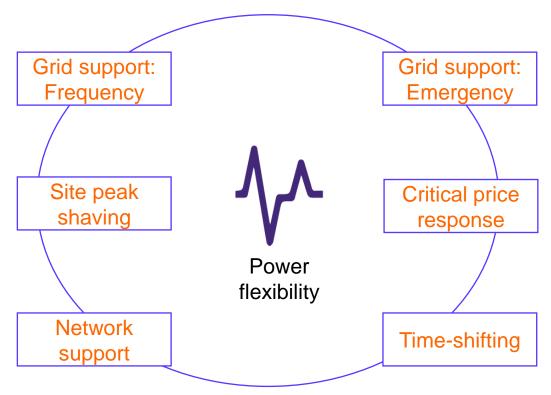
... If the Authority believes that TP's DR programme is distorting markets for flexibility and flexibility management, then the Authority, with the Commerce Commission, could consider imposing on Transpower the same cost allocation rules that are already imposed on EDBs...

- Incentivising EDB's (pg 16)
  - is the fact that it may be burdensome to the ComCom a good enough reason not to do it? What are the perverse outcomes foreseen?
  - Could the "100% renewable target" be a requirement for EDB's to consider non-network options in addition to purely economic
  - Has Australia's DMIA and DMIS been reviewed?

Further consideration



Value-stacking is important. Location is only relevant for one of six value streams (network/transmission deferral)



Further consideration



#### **Recommendations**

- 1) Strong ring-fencing rules applied before there's an issue
- 2) Stronger incentives for EBD's to use flexibility



# enelx