

The background is a complex digital composition. It features a semi-transparent globe on the left side, showing continents in a darker blue. Overlaid on this and the rest of the image are various grid patterns of different colors and opacities, including a prominent light blue grid in the top right and a darker blue grid in the bottom left. A large, curved, semi-transparent white band stretches horizontally across the middle of the image, serving as a backdrop for the title text. In the lower right, there are faint, stylized architectural elements, possibly a bridge or a modern building structure, rendered in a light blue tone.

Platform Design (II)

II. Platform Architecture Design Strategy and Features

(Key to platform strategy) Come and Stay !!

1. Platform yard: A good platform is a place where various stakeholders can collaborate and play
2. Network effect
3. Role of Platform's Director : Win-Win/Share
4. The more business partners around the platform, the more active it is → Evolving the ecosystem
5. Open Platforms and Ecosystems: Growing and Externalizing Continuous Growth
6. Killer Content: QCDRT (Quality, Cost, Delivery, Reliability, Technology)
7. Spirituality and communication: An integrated and spiritual approach and communication
8. Respect for boundary: Protect core technology, share interface technology, prohibit invasion of partner area
9. Intermediate role: interactions among ecosystem members and consequent conflicts
10. Unexpected Fun (Serendipity)

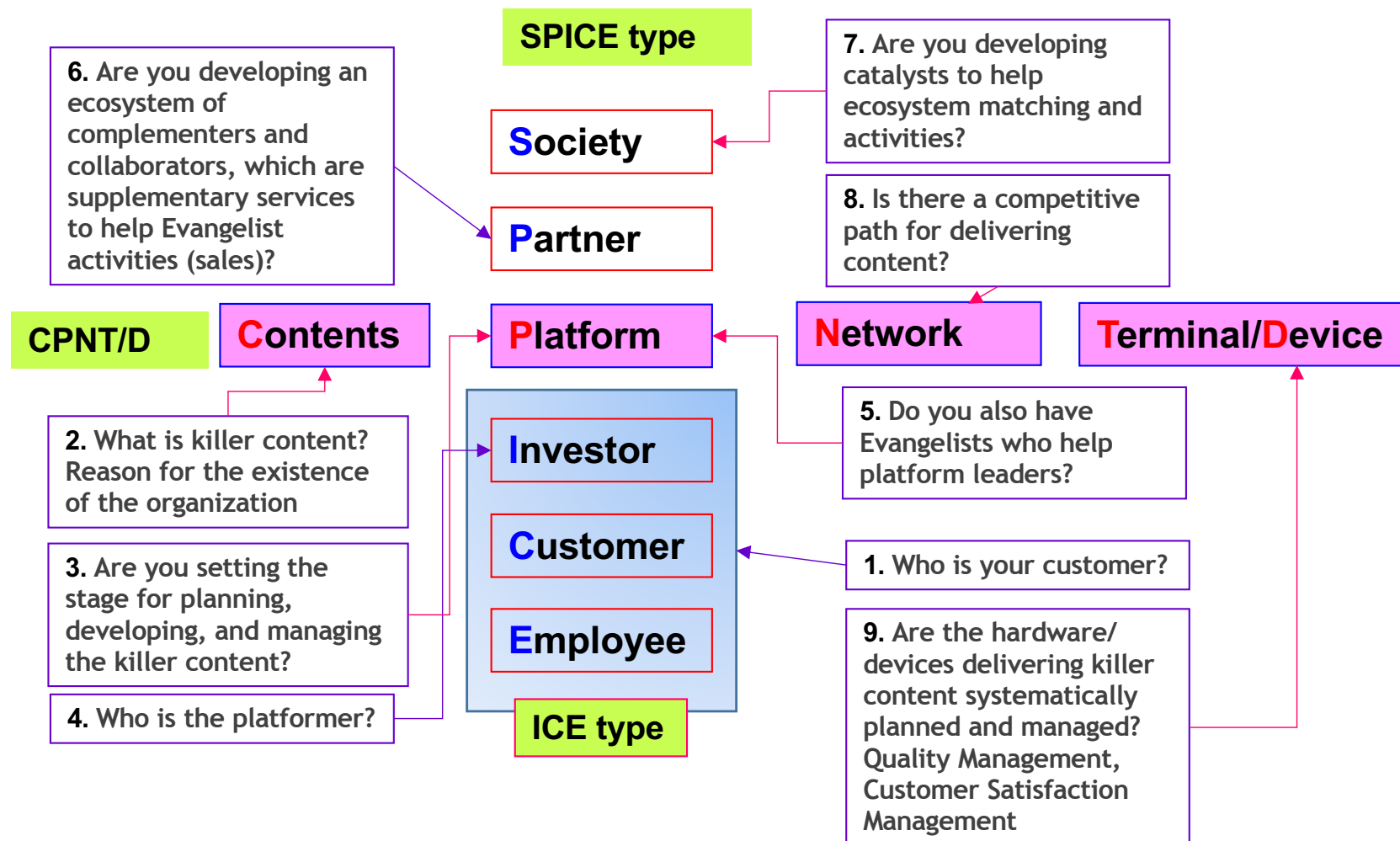
II. 플랫폼 아키텍처 설계 전략 및 특징

(플랫폼 전략의 핵심) 와서 머물게 하라 !!

1. 플랫폼 마당: 좋은 플랫폼은 다양한 이해관계자가 협력하며 뛰어 놀 수 있는 공간
2. 네트워크 효과
3. 플랫폼 지휘자 역할 : 기업간 상생/공유
4. 플랫폼 주위의 비즈니스 파트너가 많을 수록 활성화 → 생태계 진화
5. 열린 플랫폼과 열린 생태계 : 지속적인 외부성 효과를 키우고 성장
6. 킬러콘텐츠 : QCDRT (Quality, Cost, Delivery, Reliability, Technology)
7. 영성과 소통 : 통합적이고 영성적인 접근과 소통
8. 경계의 존중 : 핵심기술은 보호, 인터페이스 기술은 공유, 파트너 영역 침범 금지 등
9. 중개자 역할 : 생태계 구성원들 간의 상호작용 및 그에 따른 갈등 조정
10. 뜻밖의 재미 (Serendipity)

II. Platform Architecture Design Strategy and Features

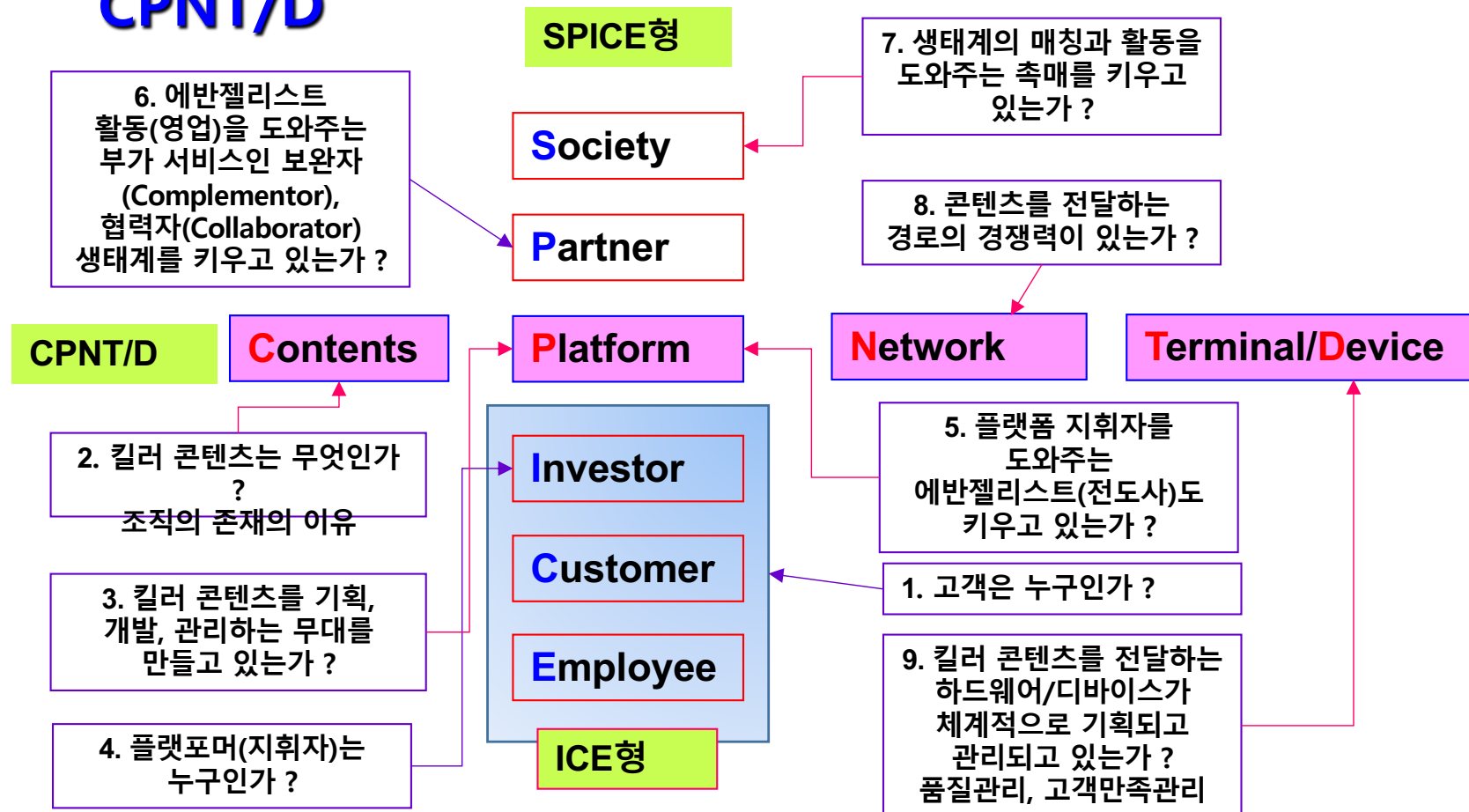
➤ Platform Architecture : SPICE ecosystem type + CPNT/D



II. Platform Architecture Design Strategy and Features

➤ 플랫폼 아키텍처 : SPICE 생태계 요소 +

CPNT/D



III. Key to Platform Success, Core/key Interactions

□ Core Interaction

- Core interactions are the most important activities happening on the platform
- When designing a platform, be the first to decide what to make core interactions
- **Participants + Value Units + Filters (SW Tool) → Core Interactions**
- LinkedIn: Experts exchange career and business strategy information, recruiters exchange job offer and candidate information, critics' opinions on global trends, etc.

❖ Excellent platform focuses on value units

- ✓ Who makes the value unit?
- ✓ How to make it and integrate it into the platform?
- ✓ Defining criteria to identify significant value units

III. 플랫폼 성패의 관건, 핵심 상호작용

□ 핵심 상호작용 (Core Interaction)

→ 핵심 상호작용은 플랫폼에서 일어나는 가장 중요한 활동

→ 플랫폼 설계시 가장 먼저 무엇을 핵심 상호작용으로 삼을지 결정

→ 참여자 + 가치 단위 + 필터 (SW 도구) → 핵심 상호작용

(예) 링크드인: 전문가들이 경력과 비즈니스 전략 정보 교환, 채용담당자들이 구인정보와 지원후보자 정보 교환, 비평가들의 글로벌 트렌드에 대한 의견 등

❖ 우수한 플랫폼은 가치 단위에 집중

→ 누가 가치 단위를 만드는 지 ?

→ 어떻게 만들어서 플랫폼에 통합 할 것인지 ?

→ 중요한 가치 단위를 분별하는 기준 정의 ?

III. Key to Platform Success, Core/key Interactions

- ✓ **Participants**
 - Clearly identify roles and scope
 - Producer + Consumer, Value Creation + Consumption
- ✓ **Value units come from core interactions**
 - The beginning of the interaction is the exchange of information
(Example) YouTube videos, Twitter tweets, LinkedIn expert profiles, Uber's available vehicle list, vehicle location, eBay product and service list, etc.
- ✓ **Filters are essential for effective use of value units**
 - Algorithm type SW-based tool used by the platform, value unit exchange/management/blocking
(Example) Matching passengers and drivers in Uber
(Example) Google's Crawler creates a web page index (value unit) → Consumer inputs a search word → Google configures a filter with other information, retweet/comment/response information → Filter is value to consumers who input search term Transfer unit

III. 플랫폼 성패의 관건, 핵심 상호작용

✓ 참여자

- 역할, 범위를 명확하게 파악
- 생산자 + 소비자, 가치창출 + 소비

✓ 가치 단위는 핵심 상호작용에서 나옴

- 상호작용의 시작은 정보의 교환

(예) 유튜브 동영상, 트위터 트윗, 링크드인 전문가 프로필, 우버의 이용 가능한 차량 목록, 차량위치, 이베이의 제품 및 서비스 목록 등

✓ 필터는 가치 단위의 효과적인 이용을 위한 필수 요소

- 플랫폼이 사용하는 알고리즘 형태의 **SW기반 도구**, 가치 단위 교환/관리/차단

(예) 우버에서 탑승객과 운전자 매칭

(예) 구글의 크롤러(Crawler)가 웹을 돌아다니며 웹페이지 인덱스(가치 단위) 생성 → 소비자가 검색어 입력 → 구글은 다른 정보, 리트윗/댓글/반응정보로 필터를 구성 → 필터는 검색어를 입력한 소비자에게 가치 단위를 전송

IV. 3 core features/functions of platform design

1. Pull (끌어오기)

- ✓ What to bring and where will you bring the producer/consumer ?
- ✓ Keep the Platform Users Staying → Encouraging Facebook's Friends (New)
- ✓ Feedback loop: a tool that keeps users coming back (expanding their activities, expanding their network, creating value, and improving network effectiveness)
- ✓ Single-user feedback loop: Analyzes user's activity/interest/preference → Recommendation
- ✓ Multi-user feedback loop: A virtuous cycle of producer-consumer activity (e.g. news feed)
- ✓ Value of tangible and intangible currencies (e.g. interest, popularity, influence, etc.)

2. Facilitate (촉진하기)

- ✓ Platforms build infrastructure to create and exchange value
- ✓ Design principles for managing interactions
- ✓ Producers make valuable commodities / services on the platform as easy as possible to exchange

3. Match (매칭하기)

- ✓ Matching properly: ensuring that relevant products and services can be traded
- ✓ Establish clear data acquisition / collection strategies / methods needed for matching

IV. 플랫폼 디자인 3대 핵심 기능

1. 끌어오기 (Pull)

- 무엇을 가지고, 누구를 (생산자/소비자) 어디서 끌어 올 것인가?
- 플랫폼 사용자들이 계속해서 머물게 → 페이스북의 친구맺기 (신규) 독려
- Feedback loop : 사용자들을 계속 돌아오게 하는 도구(자기 활동을 발생 → 네트워크 확장, 가치생성, 네트워크 효과 향상)
- Single-user feedback loop : 사용자의 활동/관심/선호도를 분석 → 추천
- Multi-user feedback loop : 생산자-소비자 활동 선순환 구조 (예) 뉴스 피드
- 유/무형 통화의 가치 (예, 관심, 인기, 영향력 등)

2. 촉진하기 (Facilitate)

- 플랫폼은 가치를 창출하고 교환하는 인프라 구축
- 상호작용을 관리하는 원칙 설계
- 생산자가 플랫폼에서 가치 있는 상품/서비스를 만들어 교환을 최대한 쉽게 해줌

3. 매칭하기 (Match)

- 제대로 매칭 : 관련성 높은 상품과 서비스를 거래할 수 있게 보장
- 매칭에 필요한 명확한 데이터 확보/수집 전략/방법 수립