

Functional Composition in Future Network Architectures

Functional composition

Current Internet

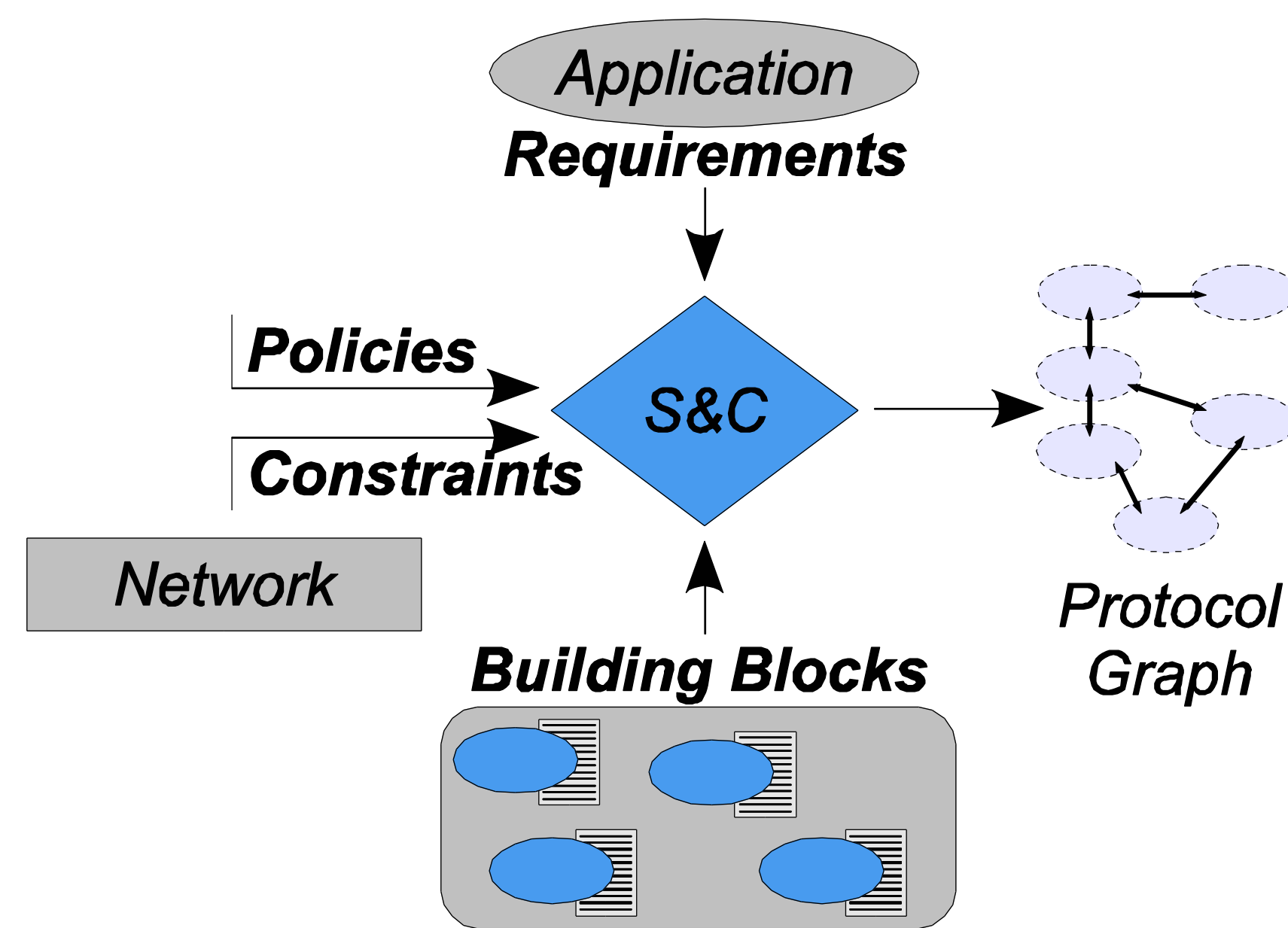
- ▶ Static dependencies of protocols
- ▶ Cannot exchange functionality easily
- ▶ **Architectural issue !**

SOA approach

- ▶ Treat network functionality as a service
- ▶ Network services are self-describing
- ▶ More flexibility, no static dependencies

Broker builds composed communication service for an application

- ▶ Application requirements + building blocks + policies & constraints = customized protocol graph
- ▶ All building block functionalities and dependencies must be compatible to each other.



Building block

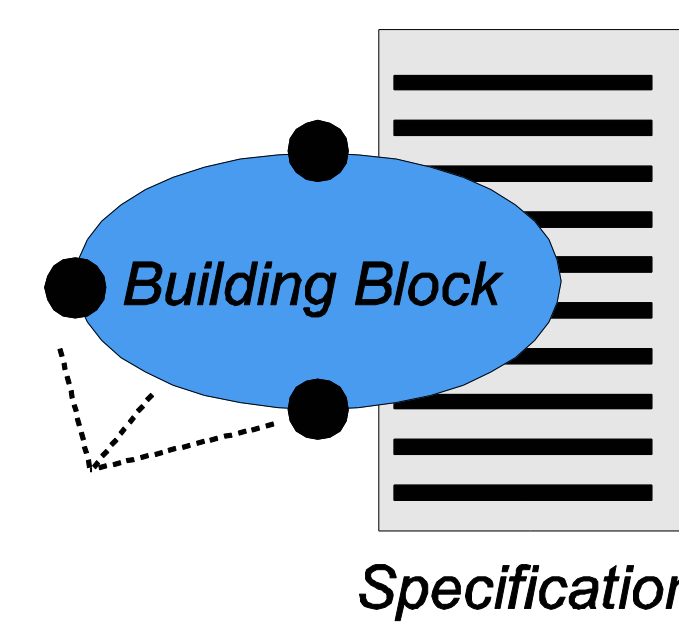
- ▶ Represents an implementation of a service
- ▶ Description of building block functionality
- ▶ Description of building block dependencies

Service

- ▶ Can be seen as a set of visible effects of the underlying implementation of a protocol or mechanism
- ▶ Hides implementation mechanism
- ▶ Examples: reliable transmission, addressing, routing, loss detection and loss reduction

Specification

- ▶ For describing offered services
- ▶ Used for comparing requirements and offerings



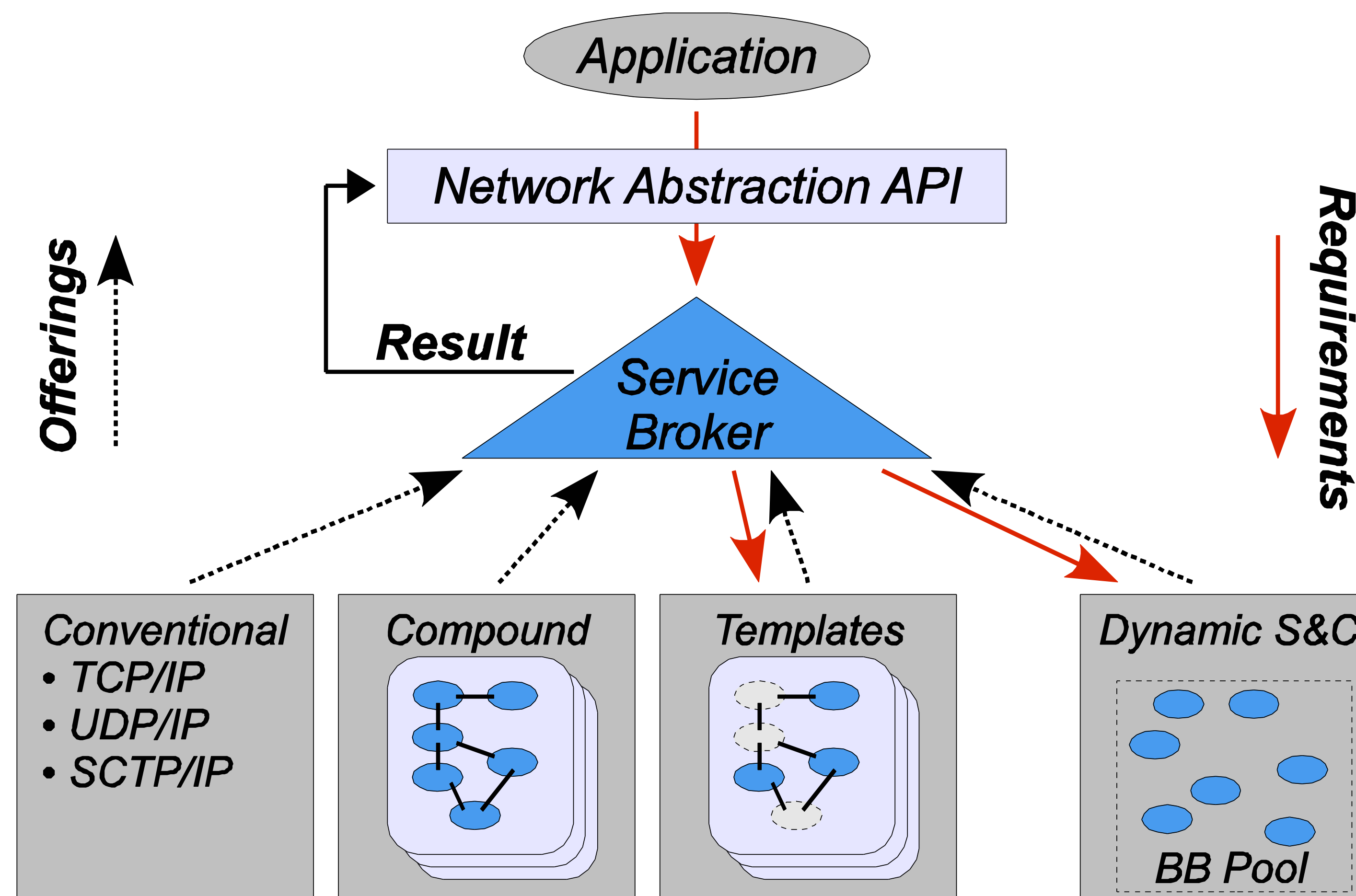
Functional Composition Approaches

Conventional

- ▶ Services provided by current protocol stacks
- ▶ Examples: TCP/IP, UDP/IP and SCTP/IP

Compound

- ▶ Design-time composition
- ▶ Tools are used to perform composition
- ▶ Trade-off between complexity and flexibility
- ▶ Less time critical



Templates

- ▶ Semi-automatic
- ▶ Selection of building blocks at run-time
- ▶ Decrease complexity by pre-defined workflow
- ▶ Useful for widely or frequently used workflows

Dynamic S&C

- ▶ Run-time composition
- ▶ Time critical
- ▶ Flexible

Challenges and Open Issues

Building Block Description

- ▶ Generic enough to accommodate future advancements
- ▶ Semantic description for automatic composition

Requirements Description

- ▶ Must be understood by different functional composition methods
- ▶ Description of qualitative and quantitative requirements
- ▶ Description of optional and mandatory requirements

Service Dependencies

- ▶ Inclusion and exclusion of a service
- ▶ Conflict among services
- ▶ Performance drain dependencies

Service Granularity

- ▶ Generic granularity
- ▶ Granularity effects dependencies among services
- ▶ Trade-off between granularity and complexity of functional composition

Functional Composition Methods

- ▶ Selection of a composition method with respect to application/domain requirements
- ▶ Trade-off between available information and complexity

Service Rating

- ▶ Rating methods
- ▶ Selection of suitable rated service
- ▶ Comparison criteria