

w e l c o m e



# Denver Dev Day – November 17, 2017







NEODESIC  
REDUX

MAINTAINABLE  
JAVASCRIPT  
APPLICATIONS

# Q U I C K I N T R O



**Thomas St Pierre**  
Principal Consultant, Neudesic

✉ [thomas.stpierre@neudesic.com](mailto:thomas.stpierre@neudesic.com)

## ABOUT THOMAS

- Extensive background in Application Architecture
- Used a purchased copy of Netscape Navigator Gold
- Counter Insurgency Expert and invented counter IED tools and tactics
- Strategic Partner in Technology and Digital Transformation
- SPA (Angular + React), MVC, WebAPI, NodeJS, ExpressJS, UWP, Xamarin, Unity3D, Cyber-Physical Simulators
- Based out of Neudesic PHX



# ABOUT NEUDESIC

Founded in  
**2002**

Privately held firm  
headquartered in Irvine, Ca  
founded by former  
Microsoft leaders.

## GLOBAL REACH

Neudesic has 13 locations in the US  
and 2 India based development centers  
that offers a broad range of strategic  
and technology services



## DEEP TECHNICAL PARTNERSHIPS



## BROAD STRATEGIC AND TECHNICAL OFFERINGS

Our consultants offer a broad range of  
technical expertise from cloud to  
mobile, from application architecture  
to developer operations.



# Agenda

---

## **WHAT AND WHY?**

What is REDUX, where did it come from and why do we care?

## **FUNDAMENTALS**

Core concepts of REDUX, how it works, what it works with

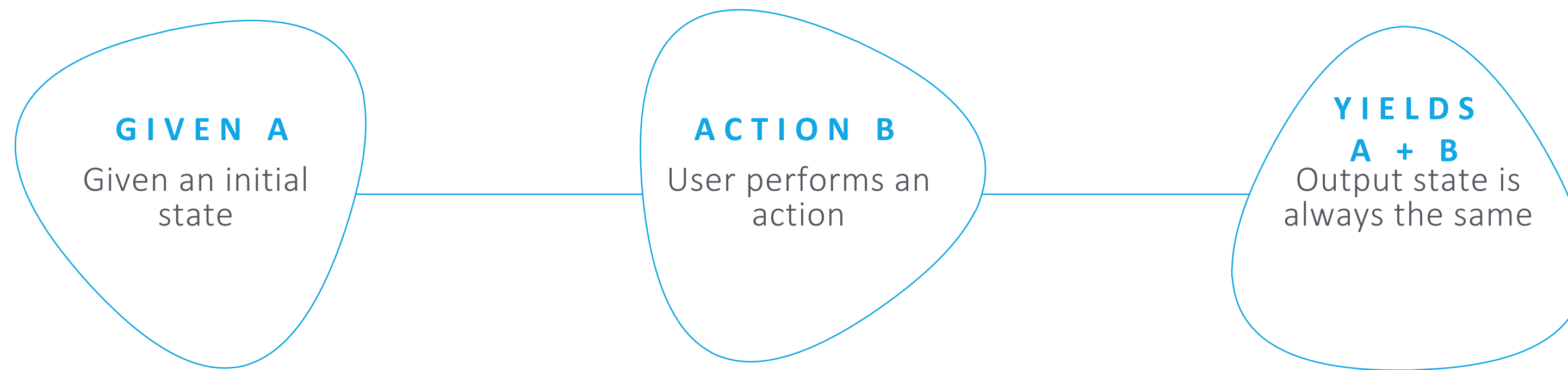
## **HANDS ON**

Explore example codebase, how it can improve dev/qa workflows and boost productivity

## **Q & A**

Open forum for any discussion/debate

WHAT IS  
**PREDICTABLE STATE**



# Predictable State

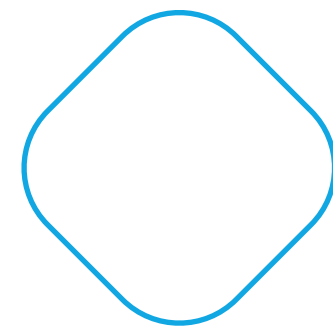
---

- User & System actions within the application always produce the same result given the same starting state
- Application cannot mutate it directly

WHAT IS  
**REDUX**

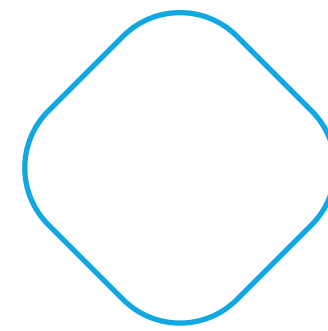


# MOTIVATIONS



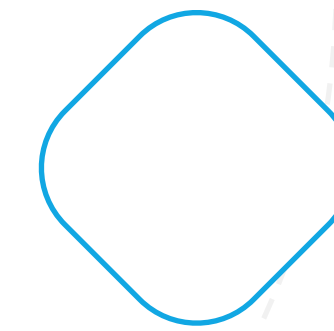
## CQRS

Query and Command model  
separation promoted One-  
Way data flow



## EVENT SOURCING

State changes driven by  
dispatching events and  
processing them semi-  
centrally



## FLUX

It's a pattern, not  
code/framework. Defines  
actions, stores, dispatchers

<https://stackoverflow.com/questions/32461229/why-use-redux-over-facebook-flux>

# THREE PRINCIPLES

---

## SINGLE SOURCE OF TRUTH

The entire application state is centralized in a single store and place.

## STATE IS READ-ONLY

State can only be changed by emitted *Actions* that describe what took place.

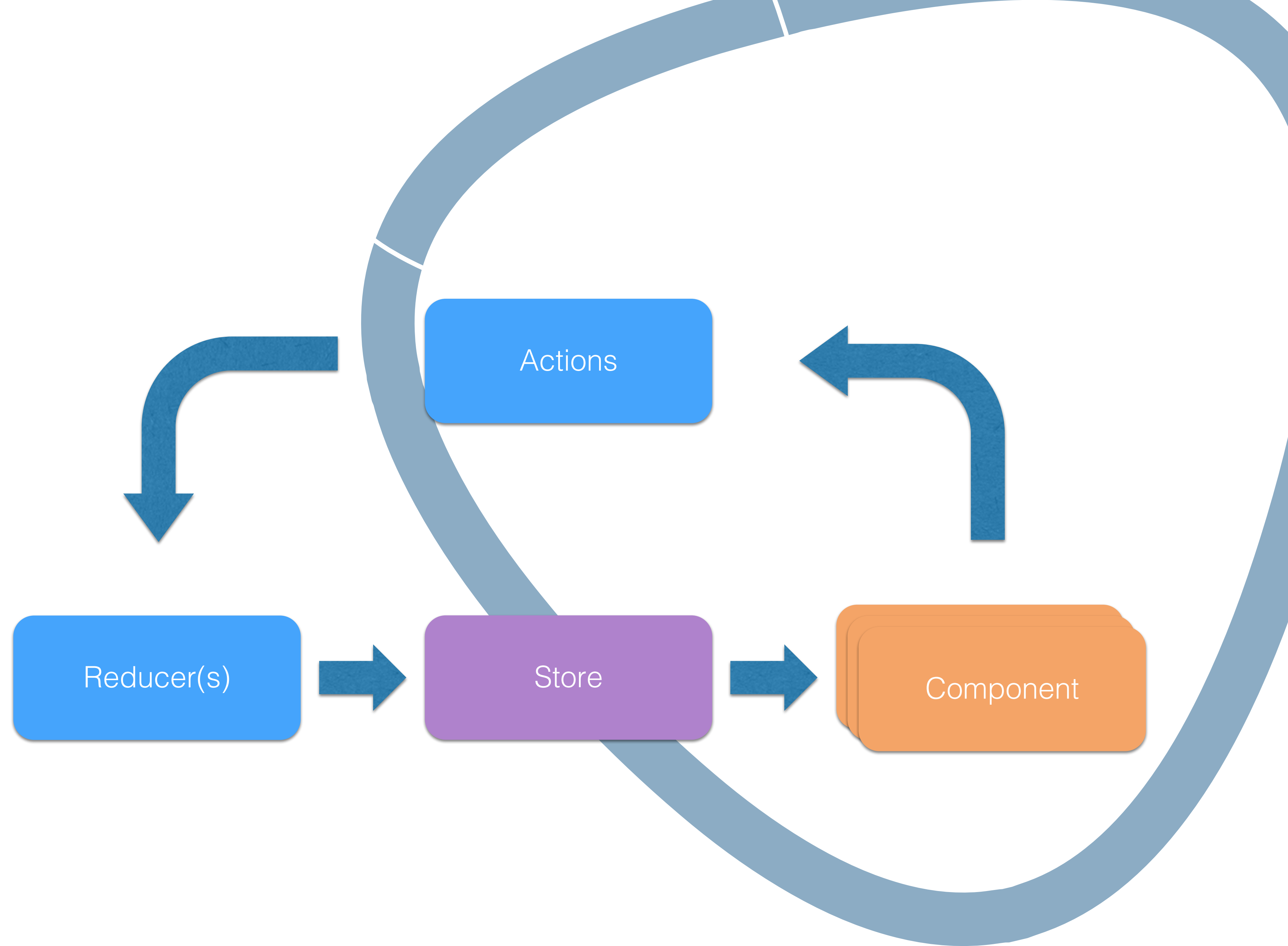
## CHANGE THROUGH PURE FUNCTIONS

Reducers should be written in pure functions agnostic of framework.

REDUX

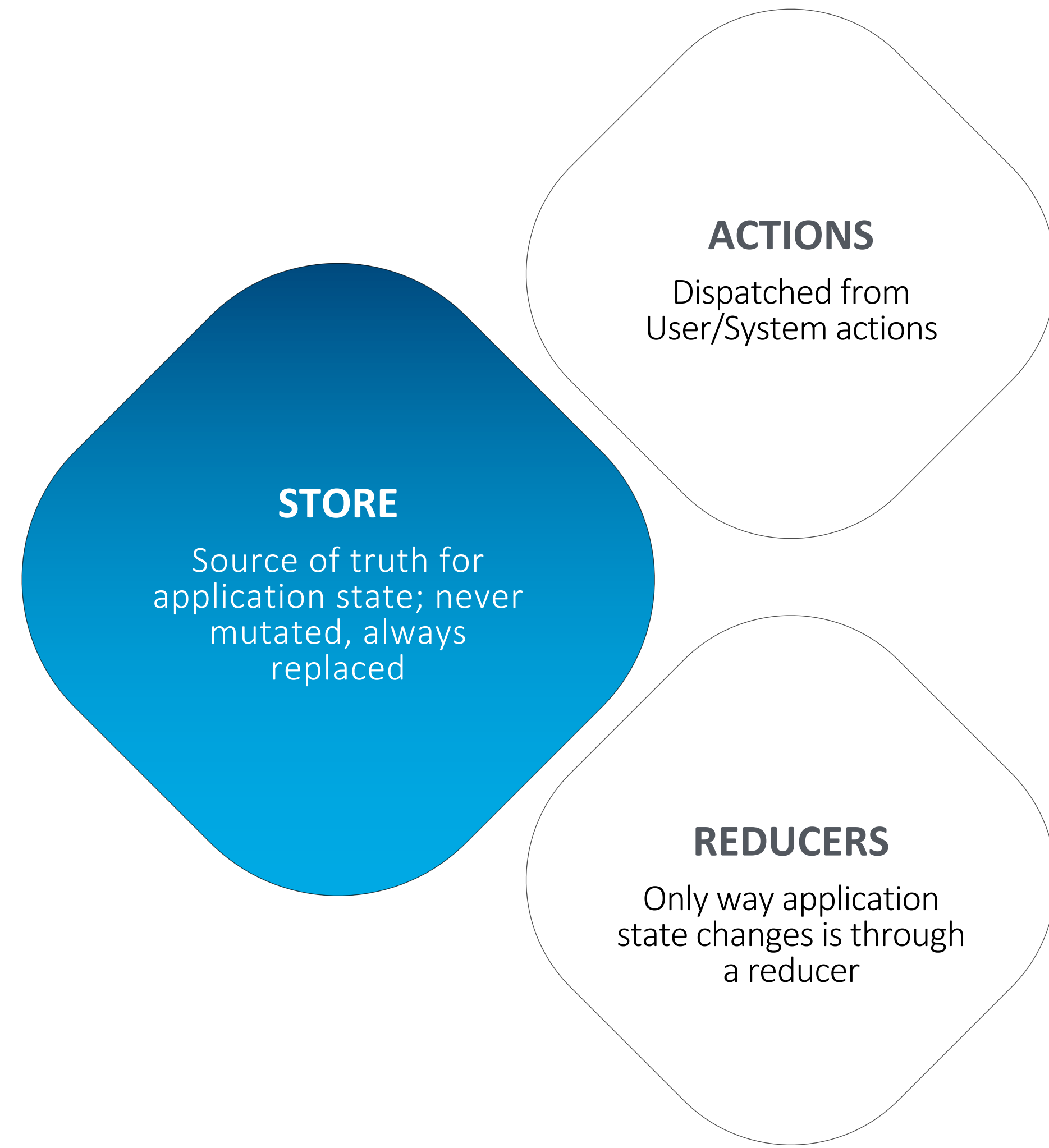
# FUNDAMENTALS

R E D U X  
C O R E



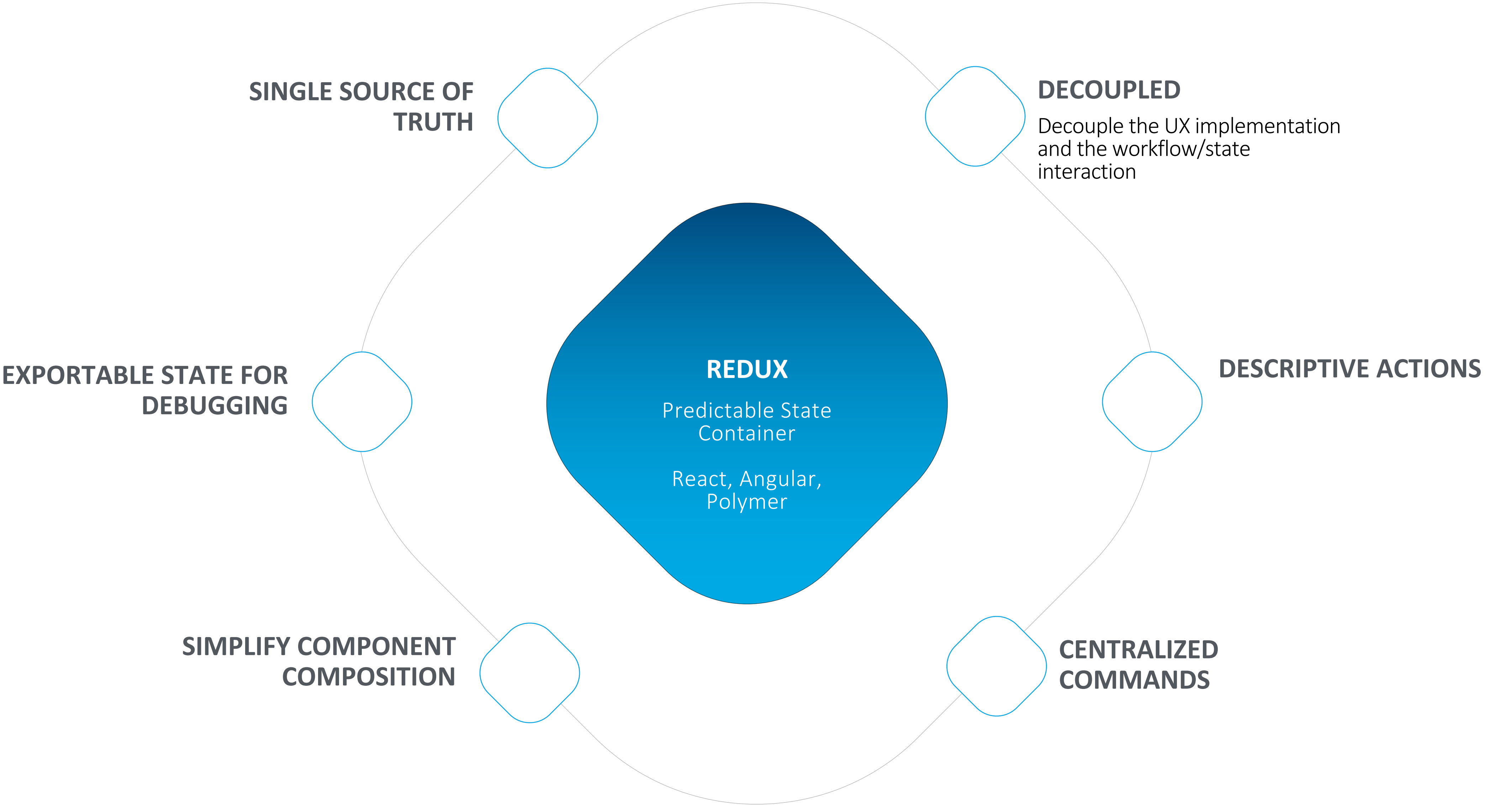


- 
- Epics help advanced Action usage (chaining Actions based on other Actions occurring)
  - Reducers can be disassembled for maintainability
  - Actions should be disassembled along with the Reducers that process them



// CODE  
**HANDS ON**

BENEFITS OF USING REDUX





THANK YOU

Contact me: [thomas.stpierre@neudesic.com](mailto:thomas.stpierre@neudesic.com)

Source code: <https://github.com/tstpierre/ddd-redux>

Shameless plug: <http://pages3.neudesic.com/Artificial-Intelligence.html>