



# End to end encryption



Data security  
from user entry to persistence data store





# Horse Racing

Before race

Race #	Ticket ID	Bet Amount	Horse	Result	Prize Amount	Create time	Update time
12	79fb4-eb o2d-3b29 b	\$3	Wagonmaster Perry			6/30/2019 11:05	

# Horse Racing

After race - superuser attack

Race #	Ticket ID	Bet Amount	Horse	Result	Prize Amount	Create time	Update time
12	79fb4-eb o2d-3b29 b	\$3	Wagonmaster Perry			6/30/2019 11:05	
			Blazing for Babes	Blazing for Babes			6/30/2019 11:37

# Horse Racing

## Prize distribution

Race #	Ticket ID	Bet Amount	Horse	Result	Prize Amount	Create time	Update time
12	79fb4-eb o2d-3b29 b	\$3	Wagonmaster Perry			6/30/2019 11:05	
			Blazing for Babes	Blazing for Babes			6/30/2019 11:37
					\$5000		6/30/2019 11:52

# Horse Racing

Event driven

Race #	Ticket ID	Bet Amount	Horse	Result	Prize Amount	Create time	Event
12	79fb4-eb o2d-3b29 b	\$3	Wagonmaster Perry	null	null	6/30/2019 11:05	Ticket purchase
12	79fb4-eb o2d-3b29 b	\$3	Wagonmaster Perry	Blazing for Babes	null	6/30/2019 11:37	Race complete
12	79fb4-eb o2d-3b29 b	\$3	Wagonmaster Perry	Blazing for Babes	\$0	6/30/2019 11:52	Prize Distribute

## Customer Browser

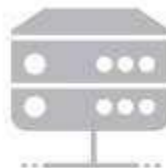


Desktop or Mobile

## Merchant Environment

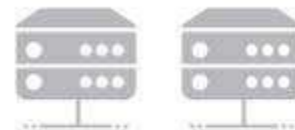


Application Systems



Back Office Systems

## Payment Processor or Gateway



SSL

SSL

Scenario 1: WITHOUT SecureData Web

Card Data Unprotected  
Full PCI scope

Scenario 2: WITH merchant-hosted SecureData Web

Card Data Protected  
Reduction in PCI scope

Scenario 3: WITH Security Voltage-enabled payment processor or gateway

Card Data Fully Protected  
Maximum reduction in PCI scope



Secure Payment Card  
Readers



Point of Sale  
(POS)



Retail Store  
IT



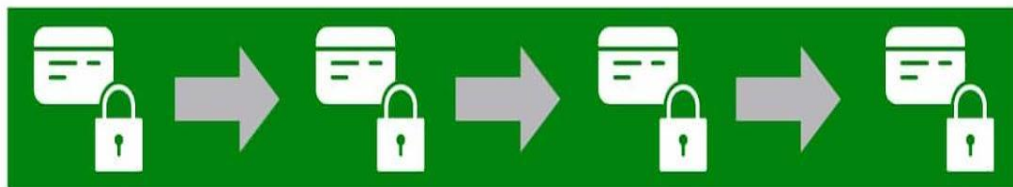
Merchant  
Switch



Merchant  
Acquirer






Issuing Bank &  
Merchant Banks



Once the card is  
processed,  
**Tokenization**  
replaces the live  
card data after  
authorization



# Voltage Format Preserving Encryption (FPE) and Tokenization

	SST	FPE		
				
	<b>Credit card</b> 1234 5678 8765 4321	<b>SSN/ID</b> 934-72-2356	<b>Email</b> bob@voltage.com	<b>DOB</b> 31-07-1966
<b>Full</b>	8736 5533 4678 9453	347-98-8309	hry@ghohawd.jiw	20-05-1972
<b>Partial</b>	<b>1234 56</b> 81 5310 <b>4321</b>	634-34- <b>2356</b>	hry@ghohawd.jiw	20-05-1972
<b>Obvious</b>	<b>1234 56</b> AZ UYTZ <b>4321</b>	AZS-UD- <b>2356</b>	hry@ghohawd.jiw	20-05-1972

Field level, format-preserving encryption, reversible data de-identification

# Kitchen counter tops



**In-Stock Countertops**



**Countertop Samples**



**Butcher Block Countertops**



**Quartz Countertops**



**Granite Countertops**



**Solid Surface Countertops**



**Laminate Countertops**



**Laminate Sheets**



**In-Stock Kitchen Cabinets**



**Kitchen Sinks**



**Kitchen Countertop Gadgets & Tools**



**Small Appliances**

# Checkout

---

<https://piefes.lowes.com/pie/v1/lowes/getkey.js>

```
// PIE version: 1.2.1
```

```
var PIE = {}; // PIE namespace
```

```
PIE.L = 6;
```

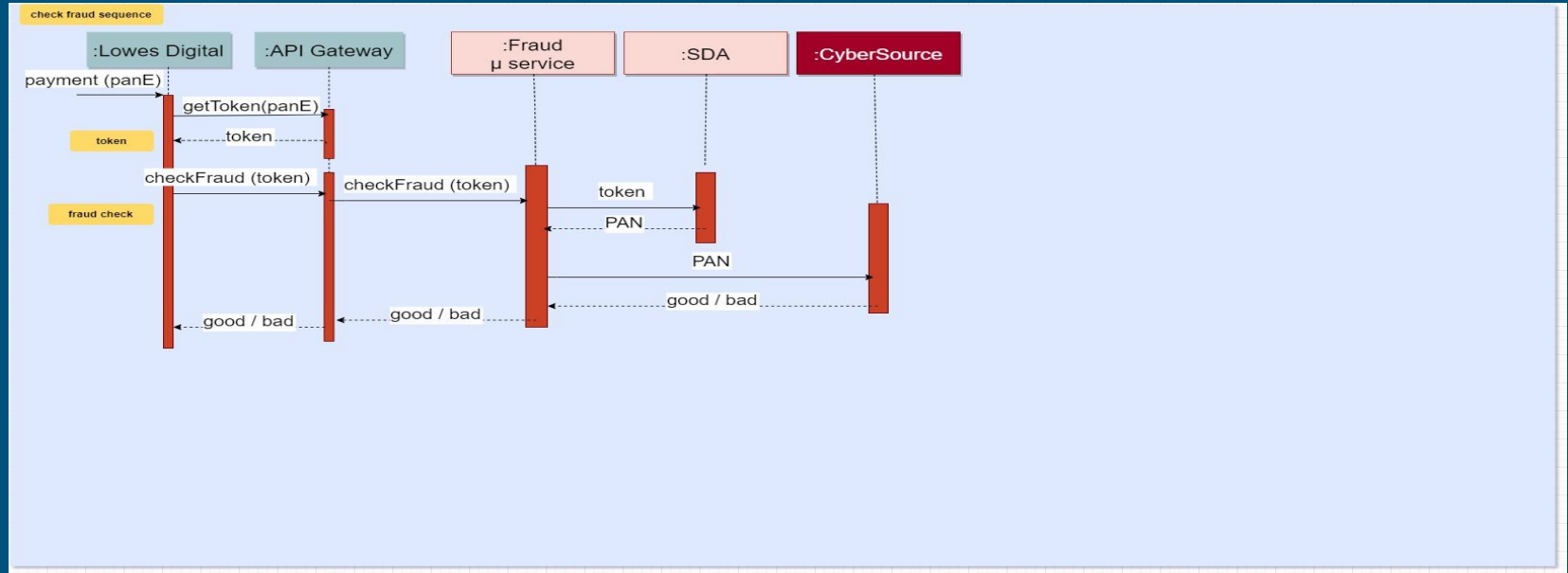
```
PIE.E = 4;
```

```
PIE.K = "04E16730F76C34DCD17B47F3124443B5";
```

```
PIE.key_id = "b2896d1d";
```

```
PIE.phase = 1;
```

# Use Token instead of PAN for transactions



# Micronaut (REST) vs gRPC

---

## Micronaut (REST) vs gRPC

```
@Get(value = "/encrypt")
public String encrypt() {
}

service SimpleService {
    rpc sayHello(HelloRequest) returns (HelloResponse);
    rpc encrypt(EncryptRequest) returns (EncryptResponse);
}
```