Toward Automated Analysis and Prototyping of Homomorphic Encryption Schemes

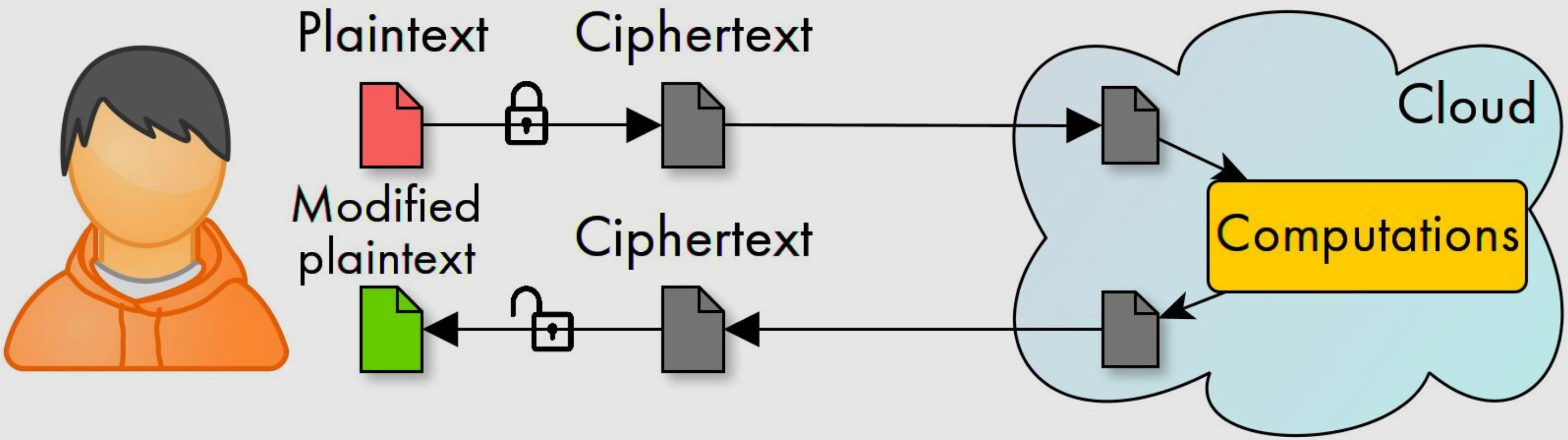
Cyrielle FERON
Loïc LAGADEC
Vianney LAPOTRE

Plan

I – Introduction

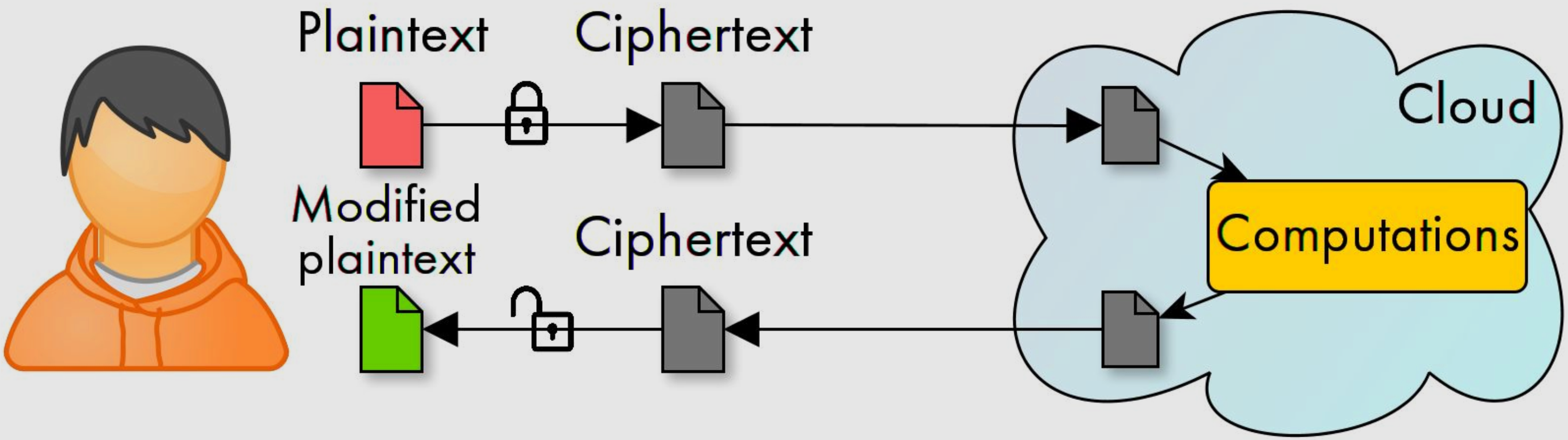
1. Homomorphic Encryption (HE)
2. Goals

Introduction – *Homomorphic Encryption (HE)*



Homomorphic Encryption principle

Introduction – *Homomorphic Encryption (HE)*

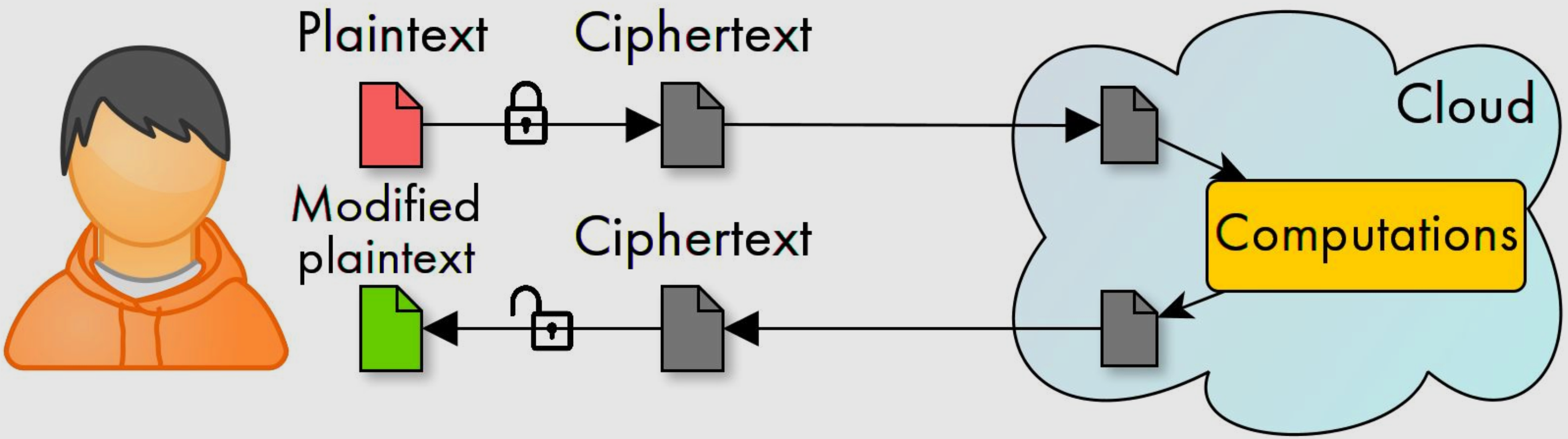


Homomorphic Encryption principle

Advantages :

- No decryption in the Cloud.
- Data are secured during the whole process (transfers and computations).

Introduction – *Homomorphic Encryption (HE)*



Homomorphic Encryption principle

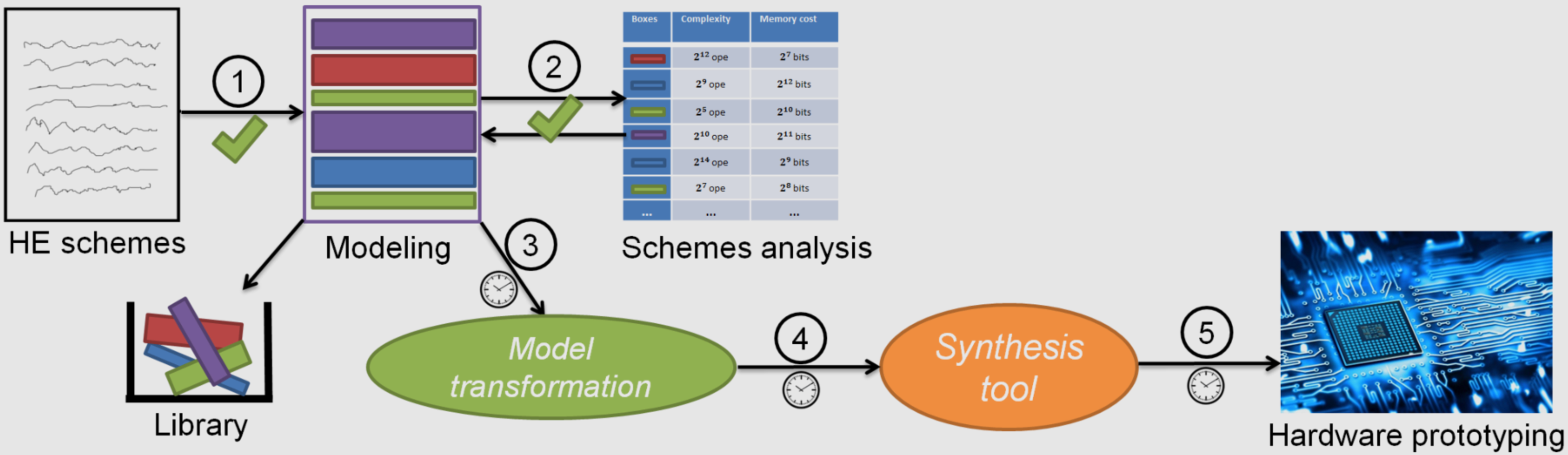
Advantages :

- No decryption in the Cloud.
- Data are secured during the whole process (transfers and computations).

Challenges :

- Active research area
  ⇨ many HE schemes.
- Important complexity and memory consumption.
- Significant expansion factor.

Develop a **Prototyping** and **Analysis Tool** for **Homomorphic Encryption Schemes (PAnTHErS)** to help improving research on HE schemes.



II – PAnTHErS modeling

1. Atomic and Specific functions
2. HE basic functions
3. Interest

Atomic functions
(= one operation)

Examples:

mult

Inputs : a, b
$c = a \times b$
Output : c

add

Inputs : a, b
$c = a + b$
Output : c

mod

Inputs : a, b
$c = a\%b$
Output : c

rand

Inputs : R, a, b
$c \leftarrow R^{a \times b}$
Output : c

PAnTHErS modeling – *Atomics and Specifics*

Atomic functions

(= one operation)

Examples:

mult

Inputs : a, b

$c = a \times b$

Output : c

add

Inputs : a, b

$c = a + b$

Output : c

mod

Inputs : a, b

$c = a\%b$

Output : c

rand

Inputs : R, a, b

$c \leftarrow R^{a \times b}$

Output : c

Specific functions

(= series of Atomic and/or Specific functions)

Example:

distriLWE

Inputs : q, n, m, k : integers
$\boldsymbol{s}$: vector of size n

$\boldsymbol{A} = rand(R_q, m, n)$

$\boldsymbol{e} = rand(\chi, m, 1)$

$\boldsymbol{e} = mult(k, \boldsymbol{e})$

$\boldsymbol{b} = mult(\boldsymbol{A}, \boldsymbol{s})$

$\boldsymbol{b} = add(\boldsymbol{e}, \boldsymbol{b})$

$\boldsymbol{b} = mod(\boldsymbol{b}, q)$

Outputs : $\boldsymbol{b}, \mathbf{A}$

PAnTHErS modeling – *HE basic functions*

HE basic functions:

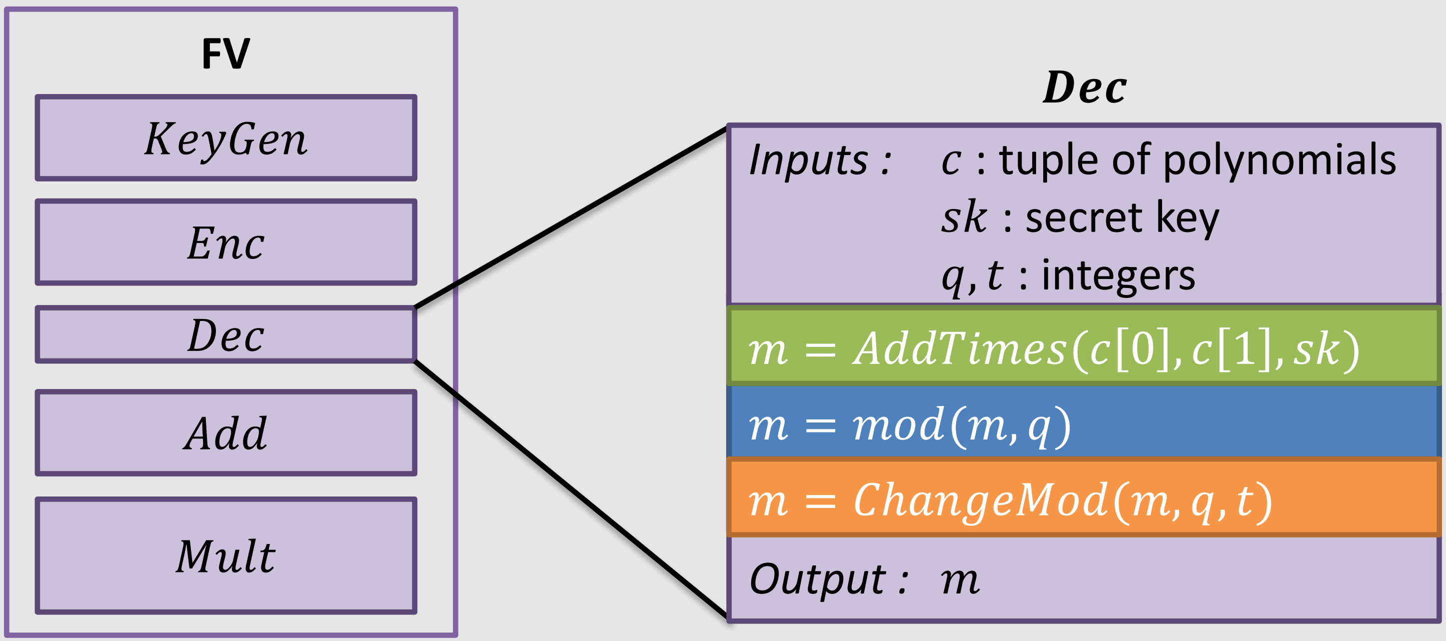
- **Series of Atomic and/or Specific functions**
- **5 HE basics: KeyGen, Enc, Dec, Add, Mult**

PAnTHErS modeling – *HE basic functions*

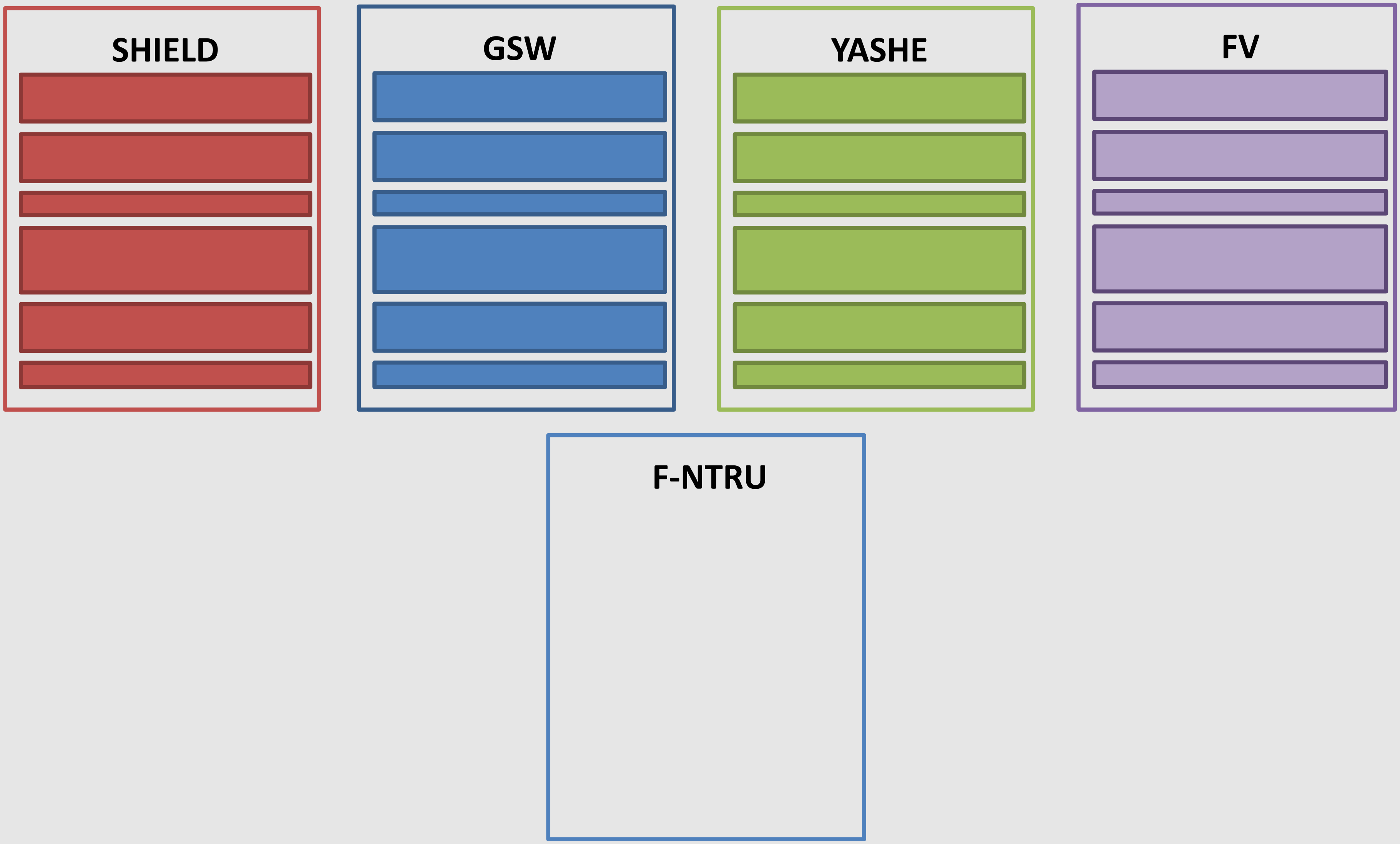
HE basic functions:

- Series of Atomic and/or Specific functions
- 5 HE basics: KeyGen, Enc, Dec, Add, Mult

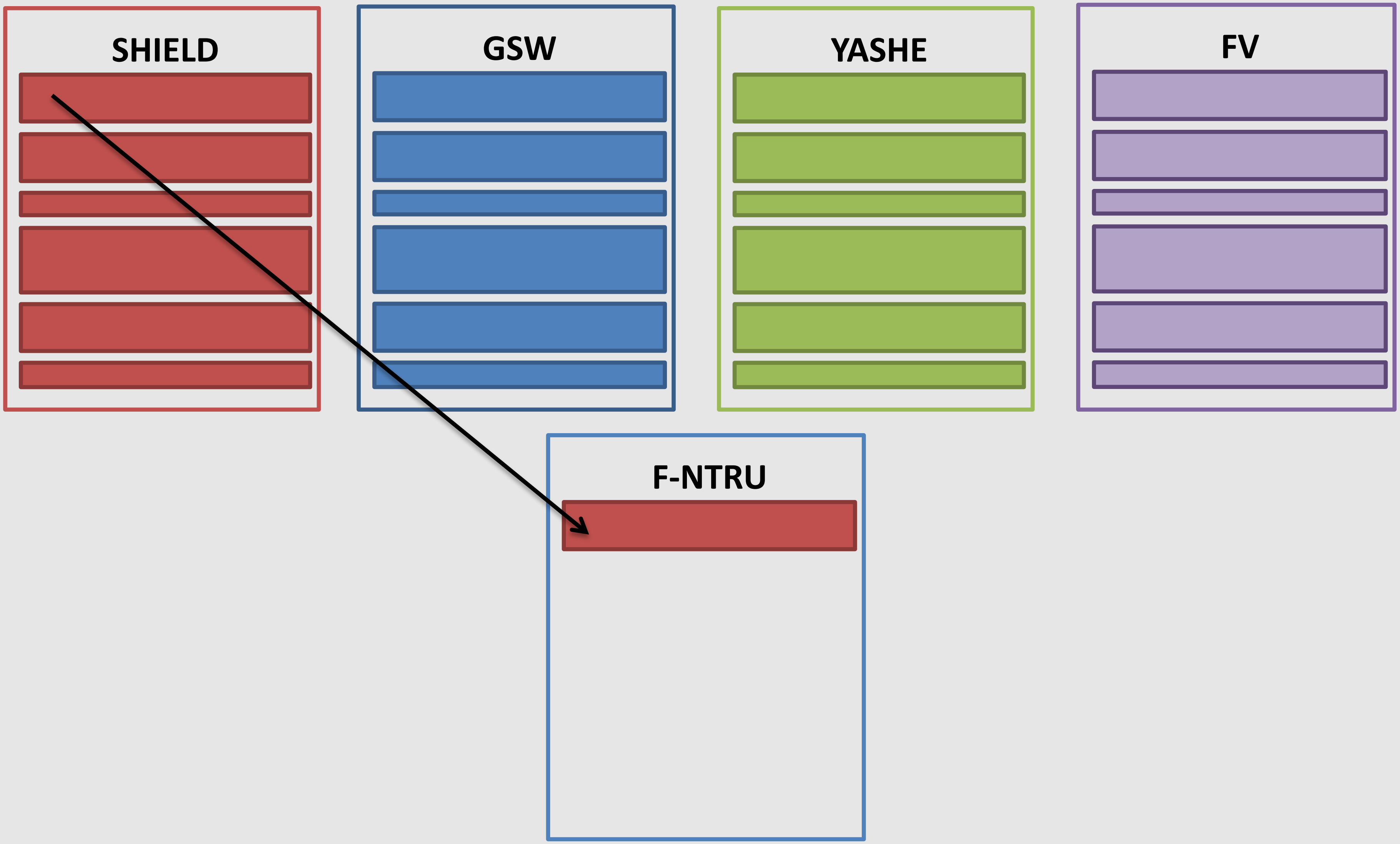
Example:



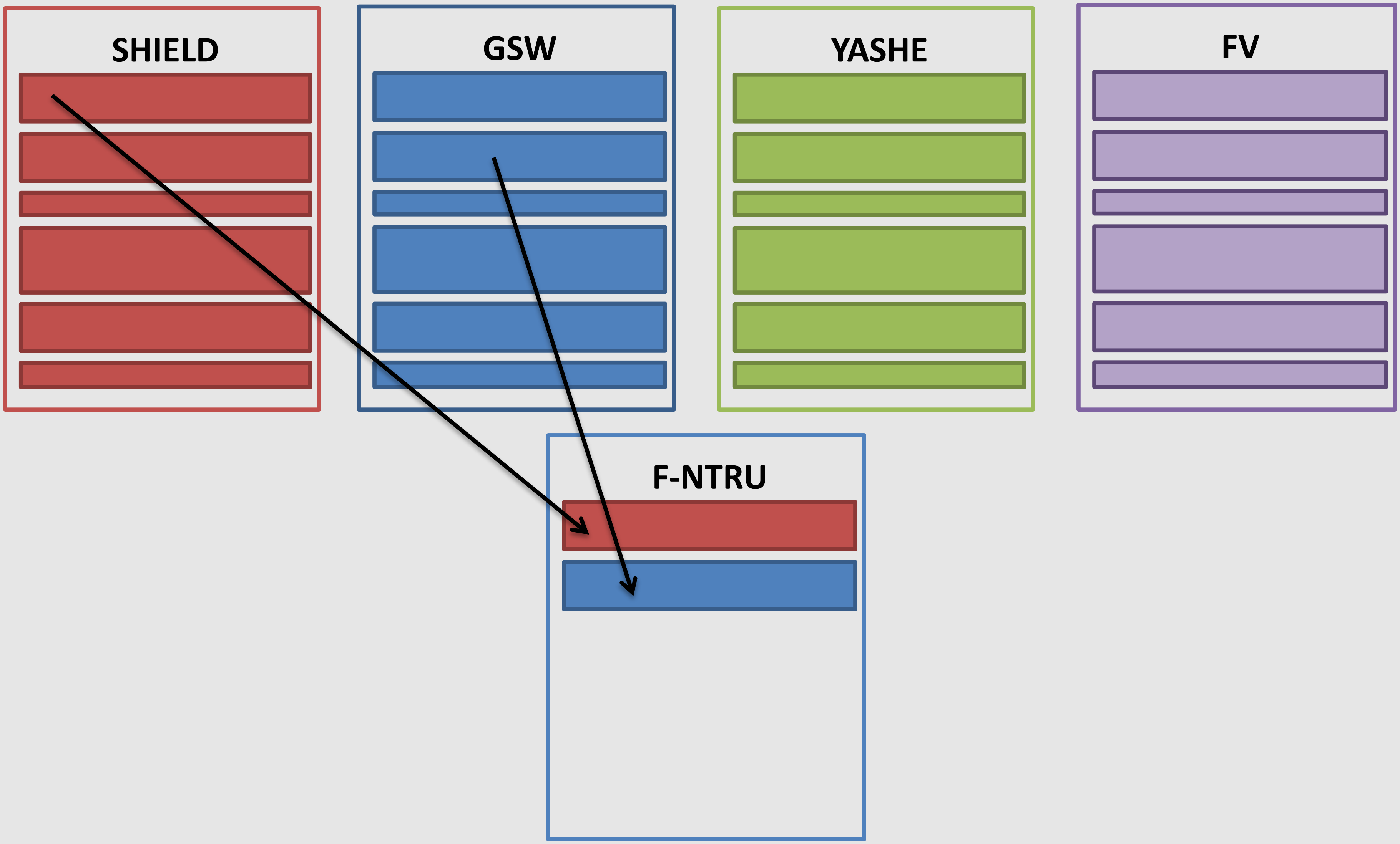
PAnTHErS modeling – *Interest*



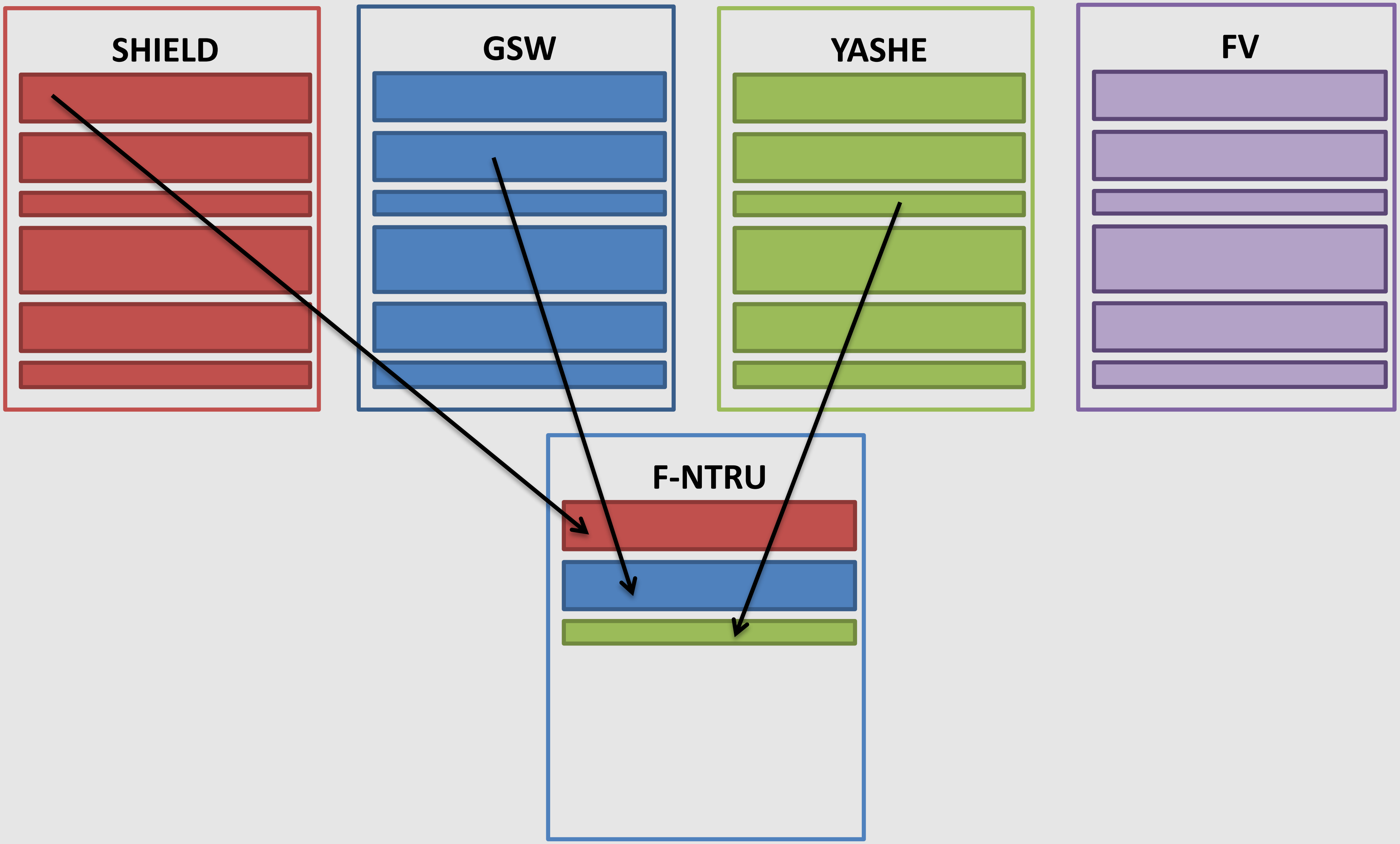
PAnTHErS modeling – *Interest*



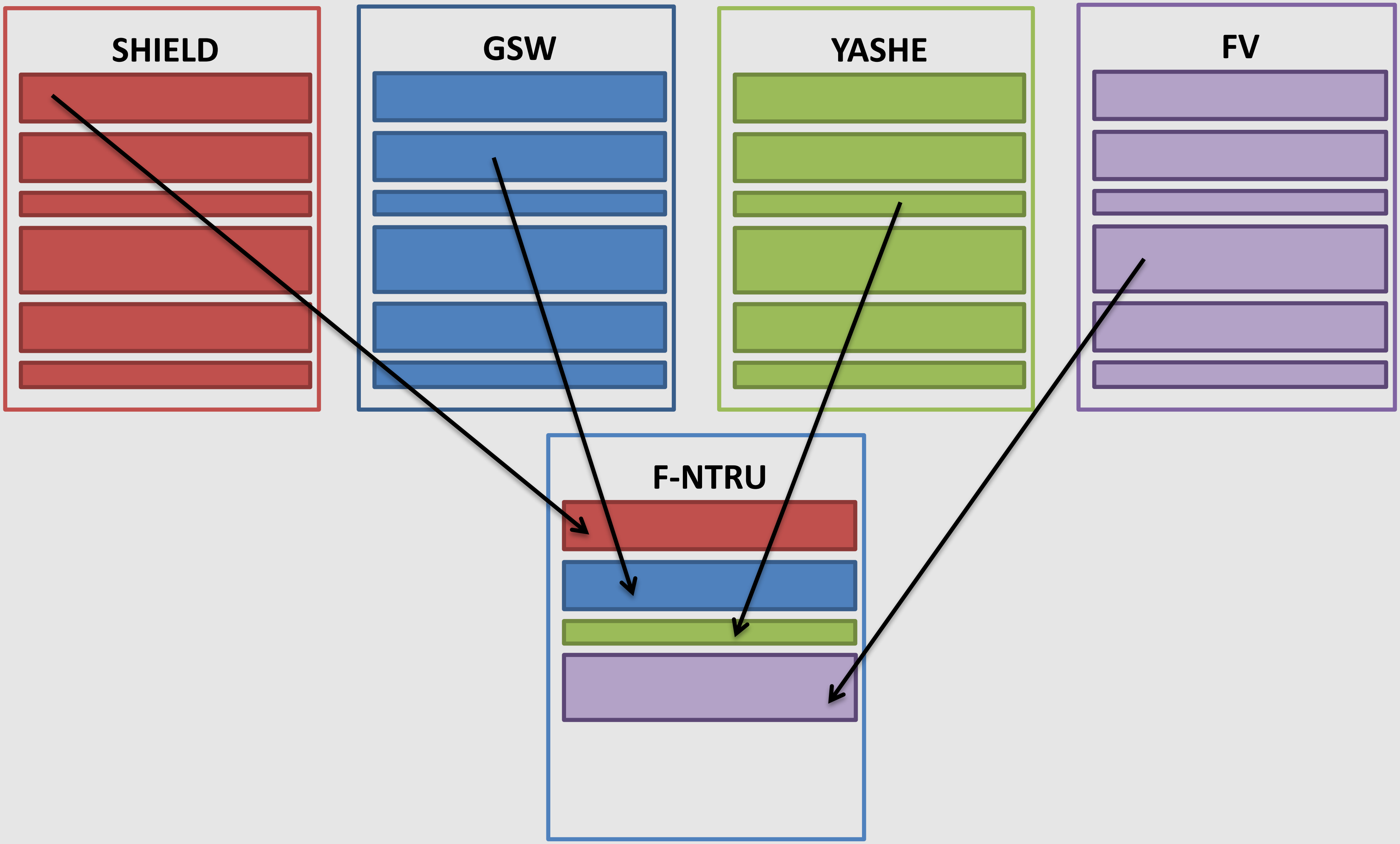
PAnTHErS modeling – *Interest*



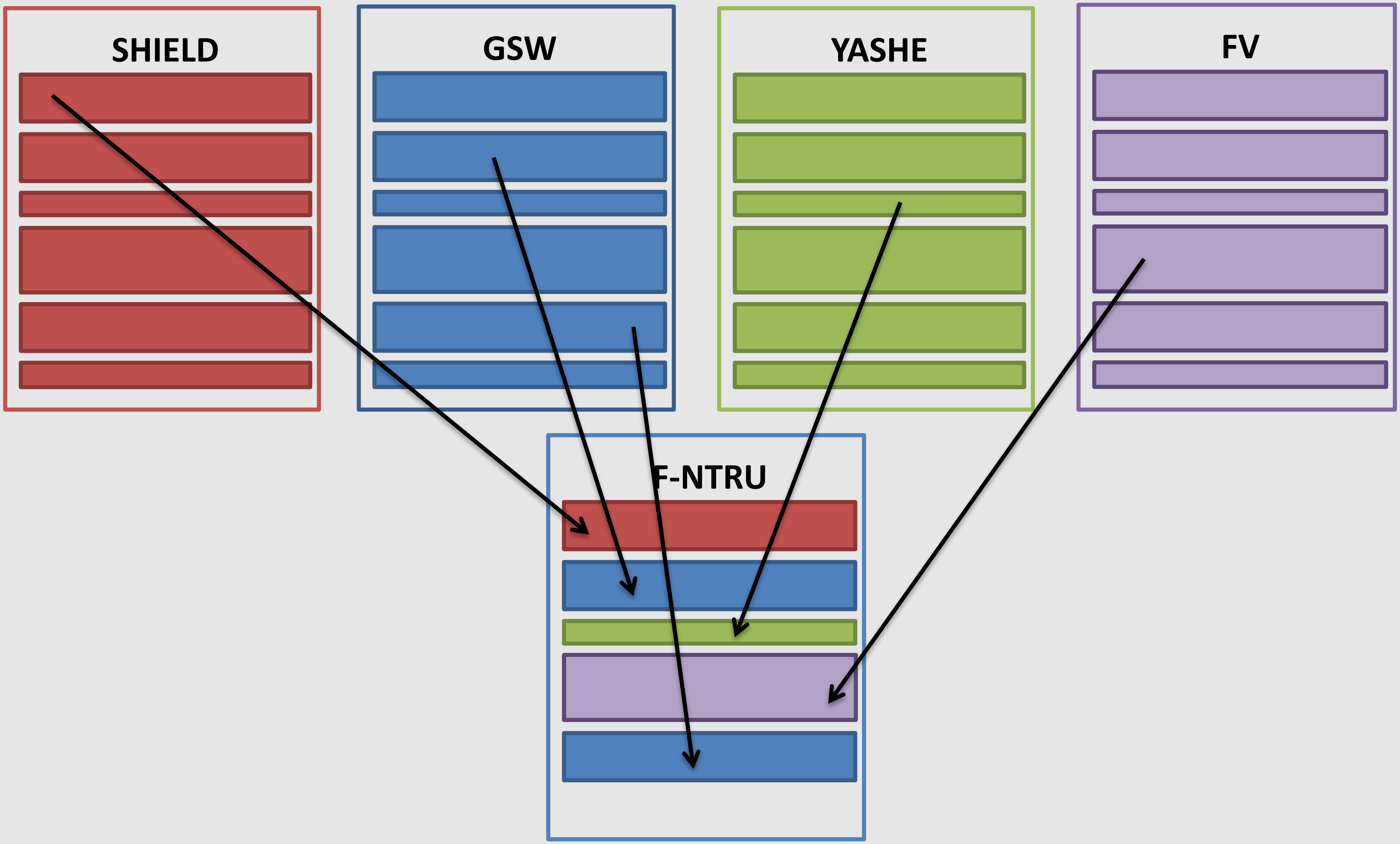
PAnTHErS modeling – *Interest*



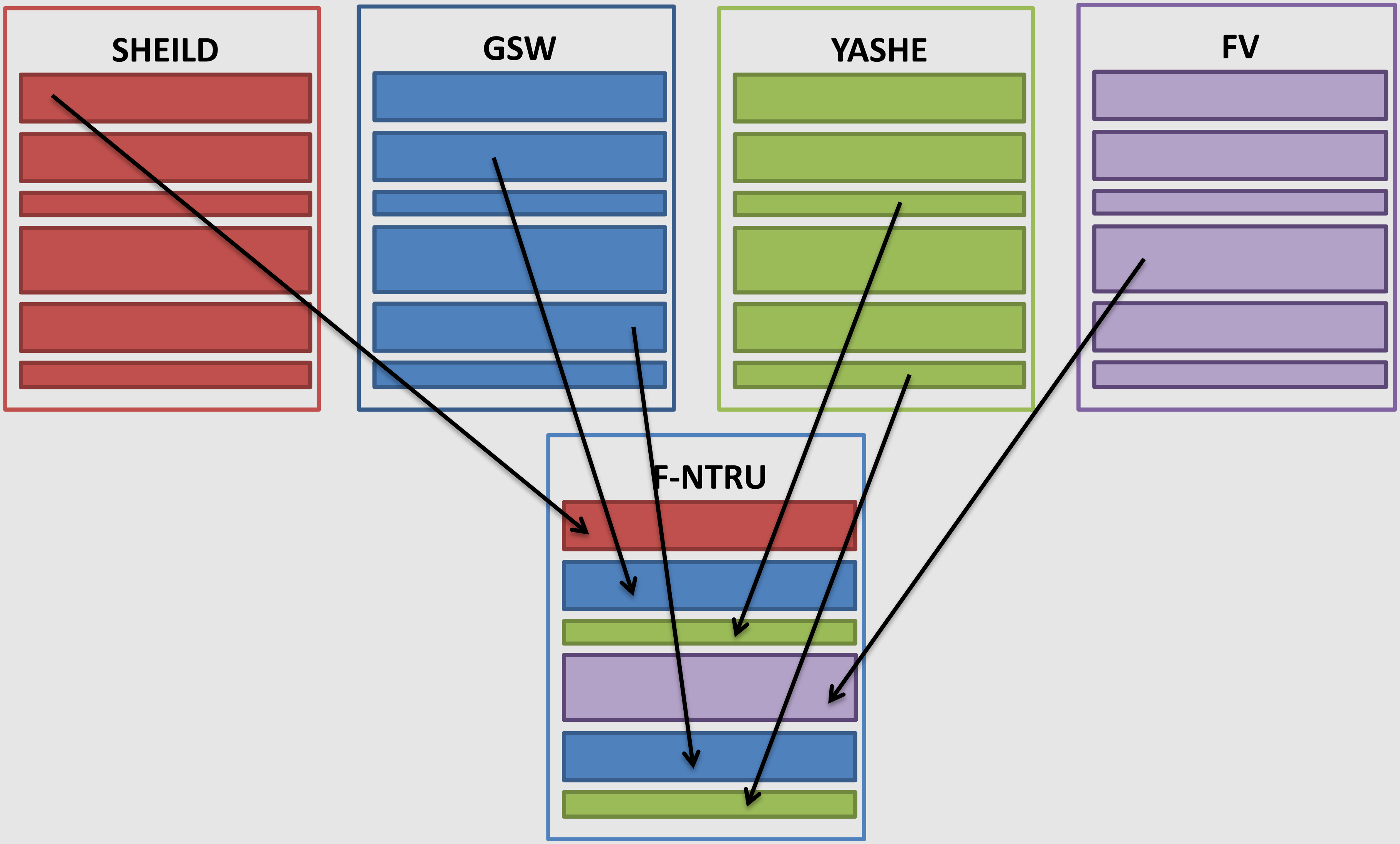
PAnTHErS modeling – *Interest*



PAnTHErS modeling – *Interest*



PAnTHErS modeling – *Interest*



III – PAnTHErS analysis

1. Complexity and memory cost
2. Atomics analysis
3. Specific and HE basic analysis
4. HE scheme analysis

PAnTHErS analysis – *Complexity and memory cost*

Complexity: table containing number of operations performed.

Example:

	MULT	ADD	DIV	MOD	RAND	ROUND
INT	$m \times d$	0	0	0	0	0
POLY	$m \times n$	$m \times n$	0	m	$(n+1).m$	0

Memory: table containing characteristics of parameters created.

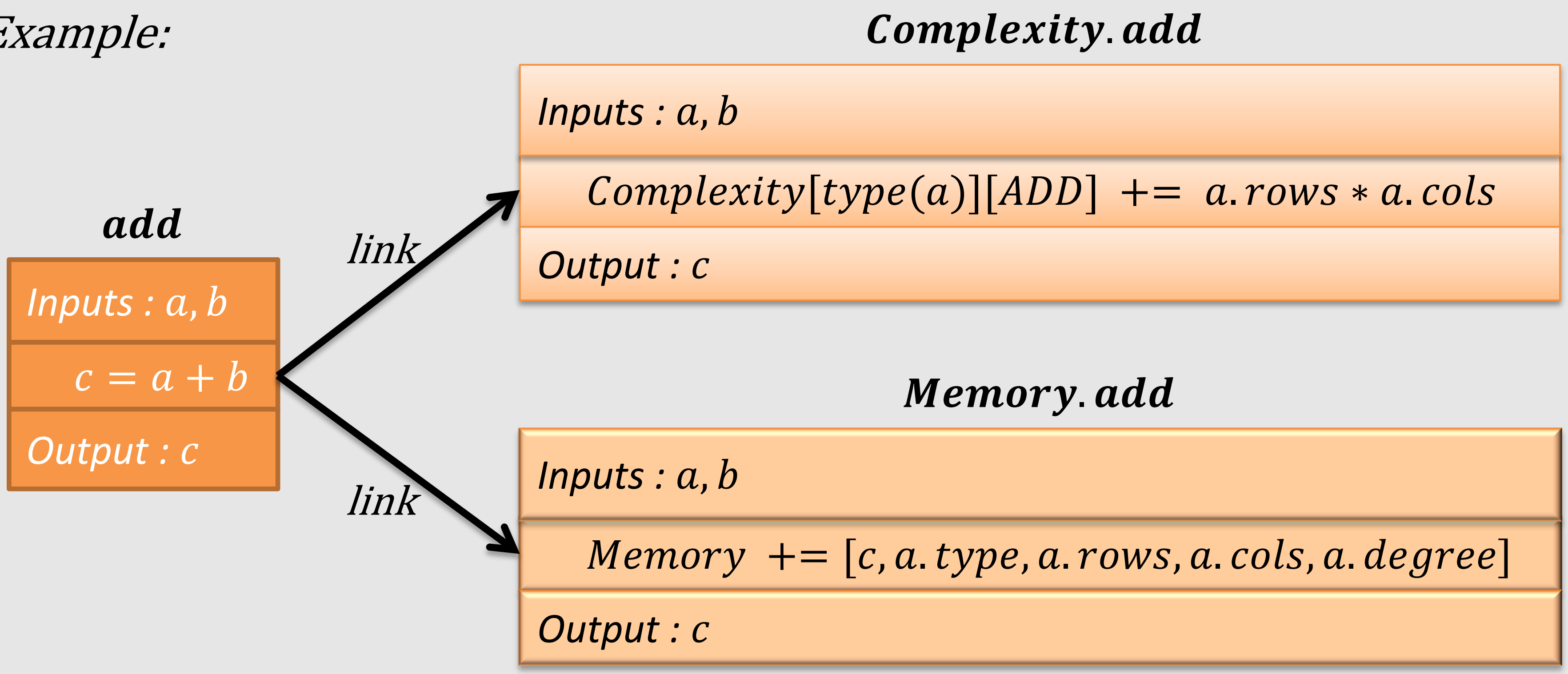
Example:

Name	Rows	Cols	Type	Degree
$\boldsymbol{A}$	n	m	POLY	2048
$\boldsymbol{c}$	m	1	POLY	2048
$\boldsymbol{b}$	m	1	POLY	2048

PAnTHErS analysis – *Atomics analysis*

Atomic functions

Example:



Specific functions

Example:

$\boldsymbol{distriLWE}$

Inputs : q, n, m, k : integers $\boldsymbol{s}$: vector of size n
$\boldsymbol{A} = rand(R_q, m, n)$
$\boldsymbol{e} = rand(\chi, m, 1)$
$\boldsymbol{e} = mult(k, \boldsymbol{e})$
$\boldsymbol{b} = mult(\boldsymbol{A}, \boldsymbol{s})$
$\boldsymbol{b} = add(\boldsymbol{e}, \boldsymbol{b})$
$\boldsymbol{b} = mod(\boldsymbol{b}, q)$
Outputs : $\boldsymbol{b}, \mathbf{A}$

Specific functions

Example:

$\boldsymbol{distriLWE}$		$\boldsymbol{Complexity.distriLWE}$
Inputs : q, n, m, k : integers $\boldsymbol{s}$: vector of size n		*Inputs :* q, n, m, k : integers $\boldsymbol{s}$: vector of size n
$\boldsymbol{A} = rand(R_q, m, n)$		$\boldsymbol{A} = Complexity.rand(R_q, m, n)$
$\boldsymbol{e} = rand(\chi, m, 1)$		$\boldsymbol{e} = Complexity.rand(\chi, m, 1)$
$\boldsymbol{e} = mult(k, \boldsymbol{e})$	⇒	$\boldsymbol{e} = Complexity.mult(k, \boldsymbol{e})$
$\boldsymbol{b} = mult(\boldsymbol{A}, \boldsymbol{s})$		$\boldsymbol{b} = Complexity.mult(\boldsymbol{A}, \boldsymbol{s})$
$\boldsymbol{b} = add(\boldsymbol{e}, \boldsymbol{b})$		$\boldsymbol{b} = Complexity.add(\boldsymbol{e}, \boldsymbol{b})$
$\boldsymbol{b} = mod(\boldsymbol{b}, q)$		$\boldsymbol{b} = Complexity.mod(\boldsymbol{b}, q)$
Outputs : $\boldsymbol{b}, \mathbf{A}$		*Outputs :* $\boldsymbol{b}, \mathbf{A}$

HE basic functions

Example:

$\boldsymbol{FVDec}$	
Inputs :	c : tuple of polynomials sk : secret key
$m = AddTimes(c[0], c[1], sk)$	
$m = mod(m, q)$	
$m = ChangeMod(m, q, t)$	
Output :	m

HE basic functions

Example:

$\boldsymbol{FVDec}$

Inputs : c : tuple of polynomials
sk : secret key

$m = AddTimes(c[0], c[1], sk)$

$m = mod(m, q)$

$m = ChangeMod(m, q, t)$

Output : m

$\Longrightarrow$

$Memory.\boldsymbol{FVDec}$

Inputs : c : tuple of polynomials
sk : secret key

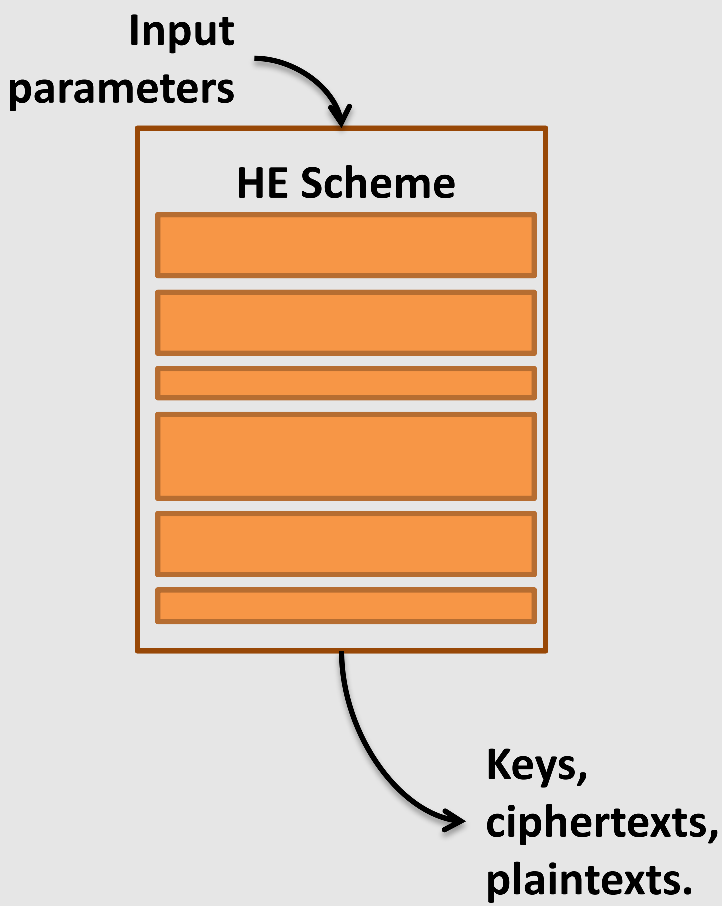
$m = Memory.AddTimes(c[0], c[1], sk)$

$m = Memory.mod(m, q)$

$m = Memory.ChangeMod(m, q, t)$

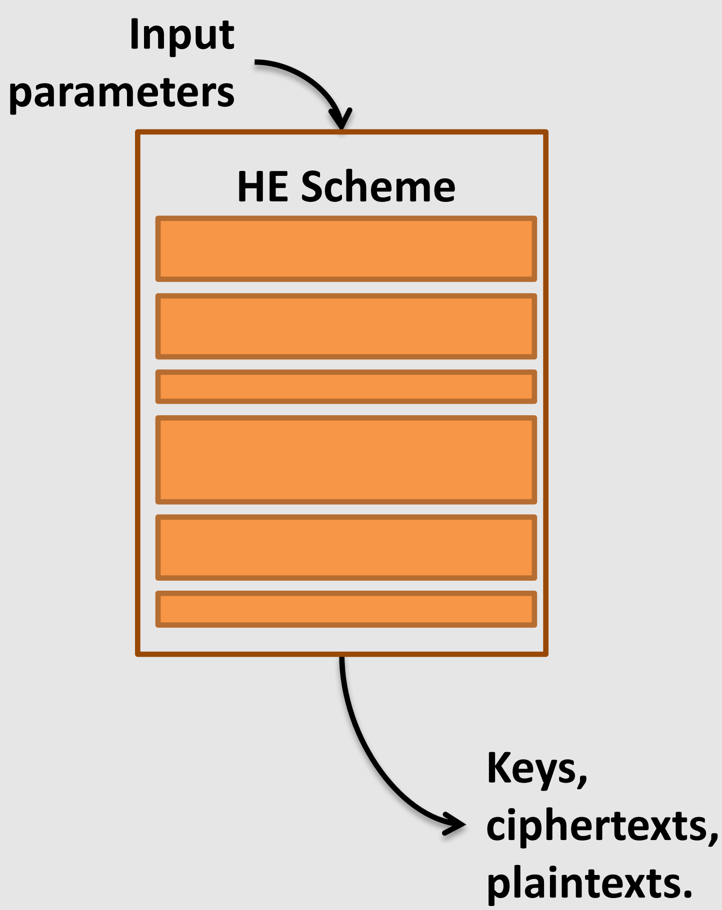
Output : m

Execution of a HE scheme

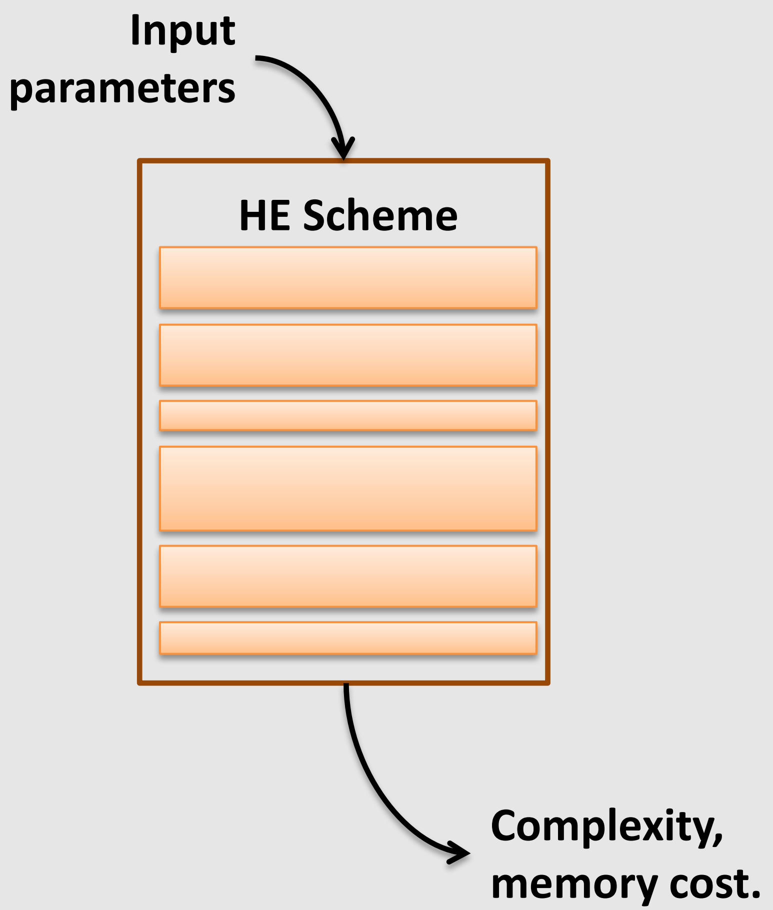


PAnTHErS analysis – *HE scheme analysis*

Execution of a HE scheme

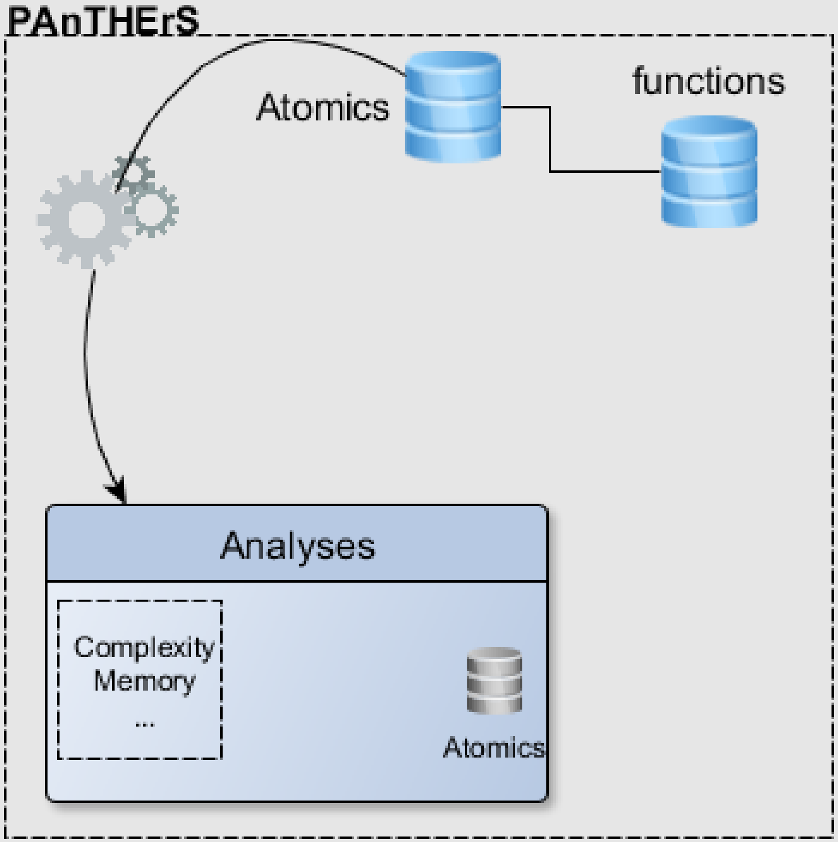


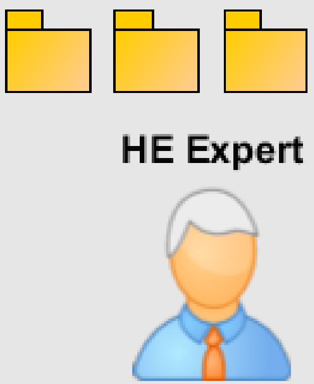
Execution of HE scheme analysis



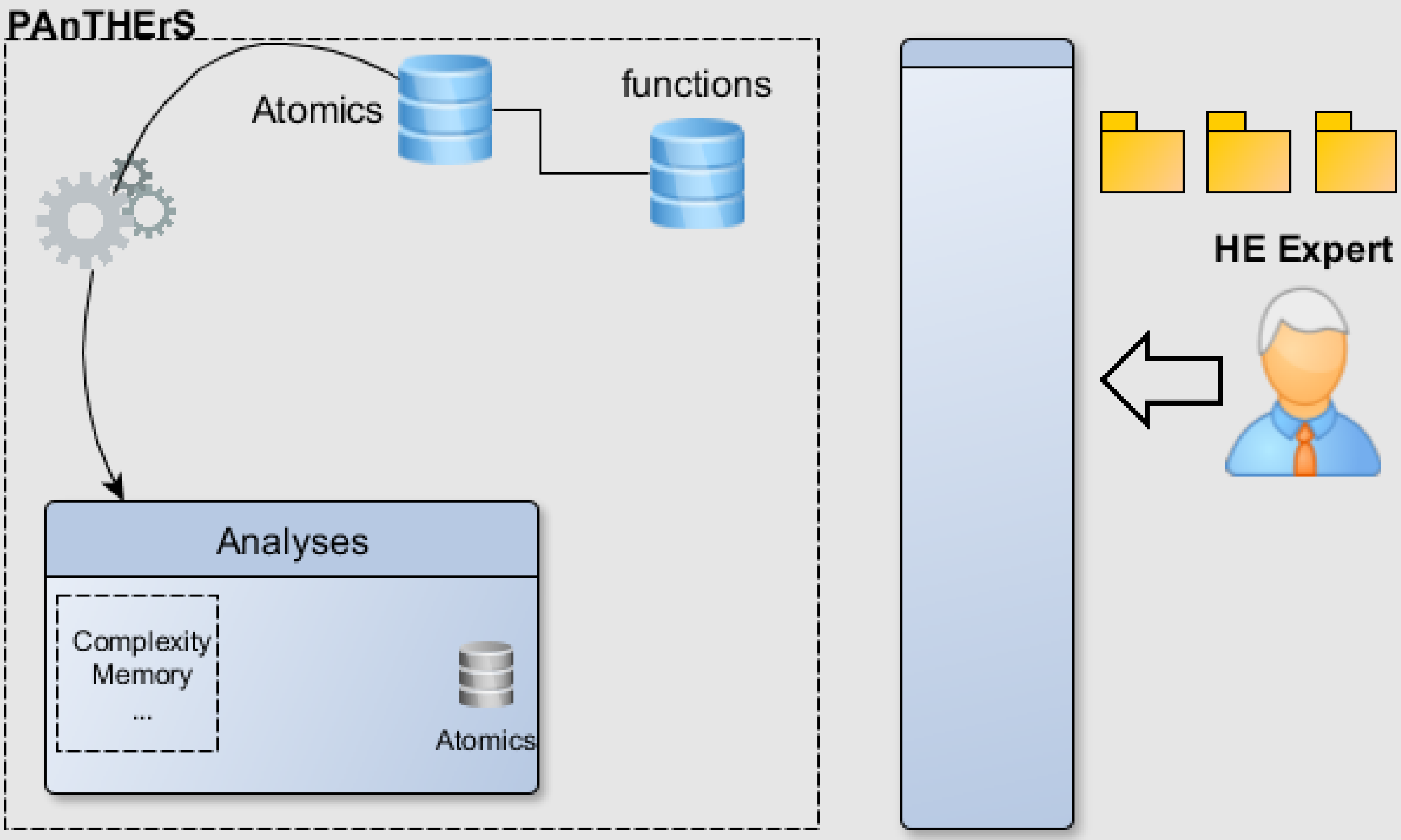
IV – PAnTHErS usage

PAnTHErS usage

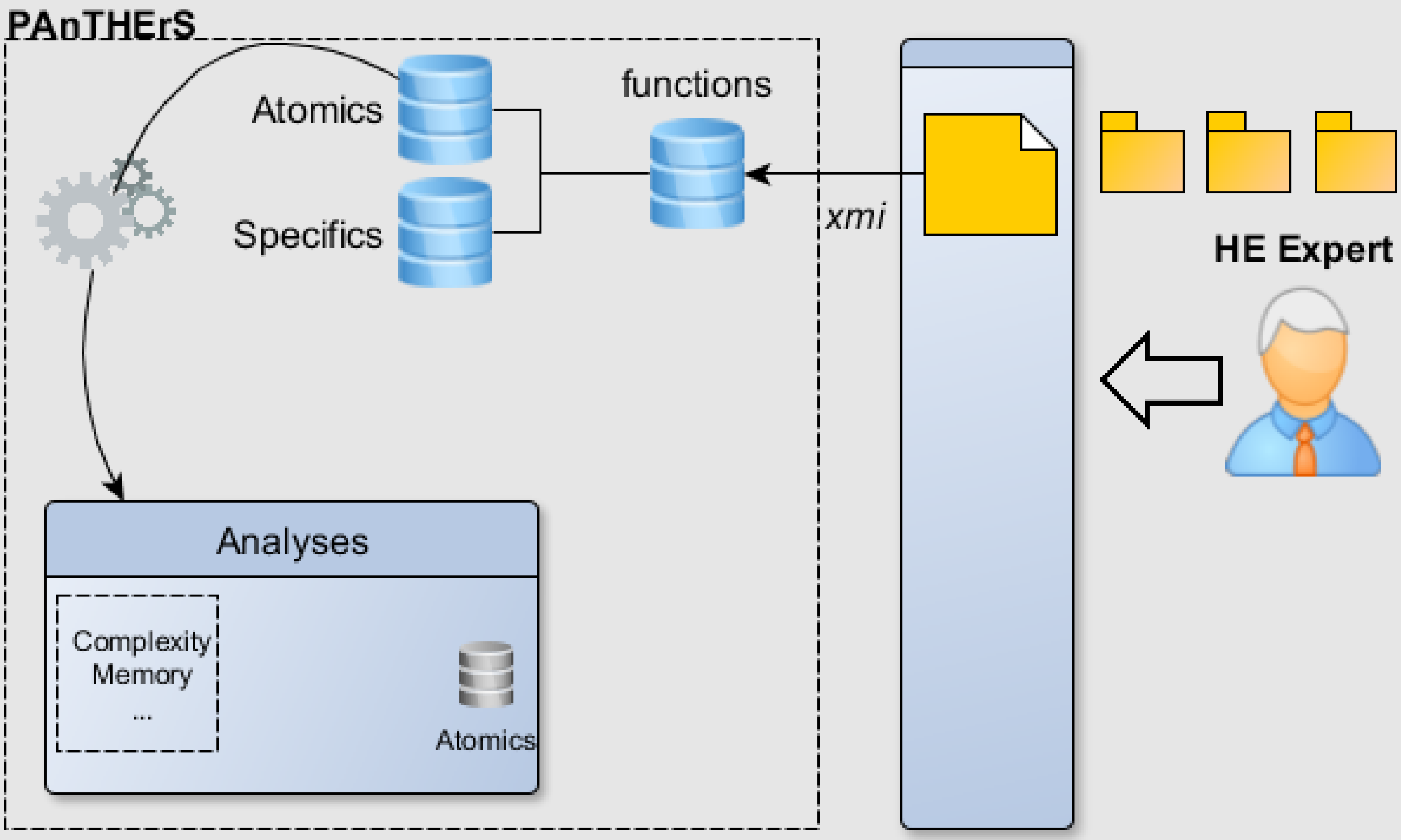




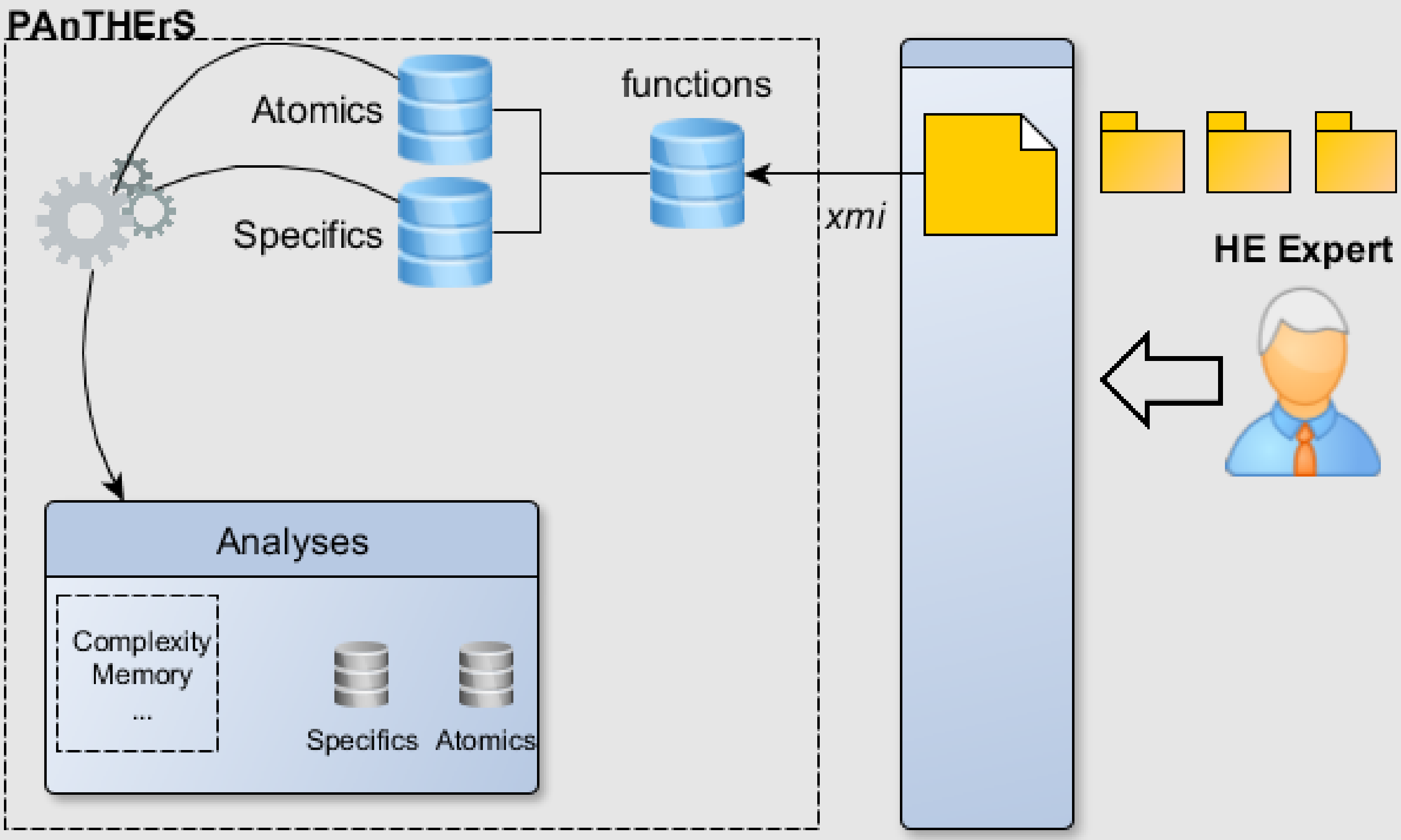
PAnTHErS usage



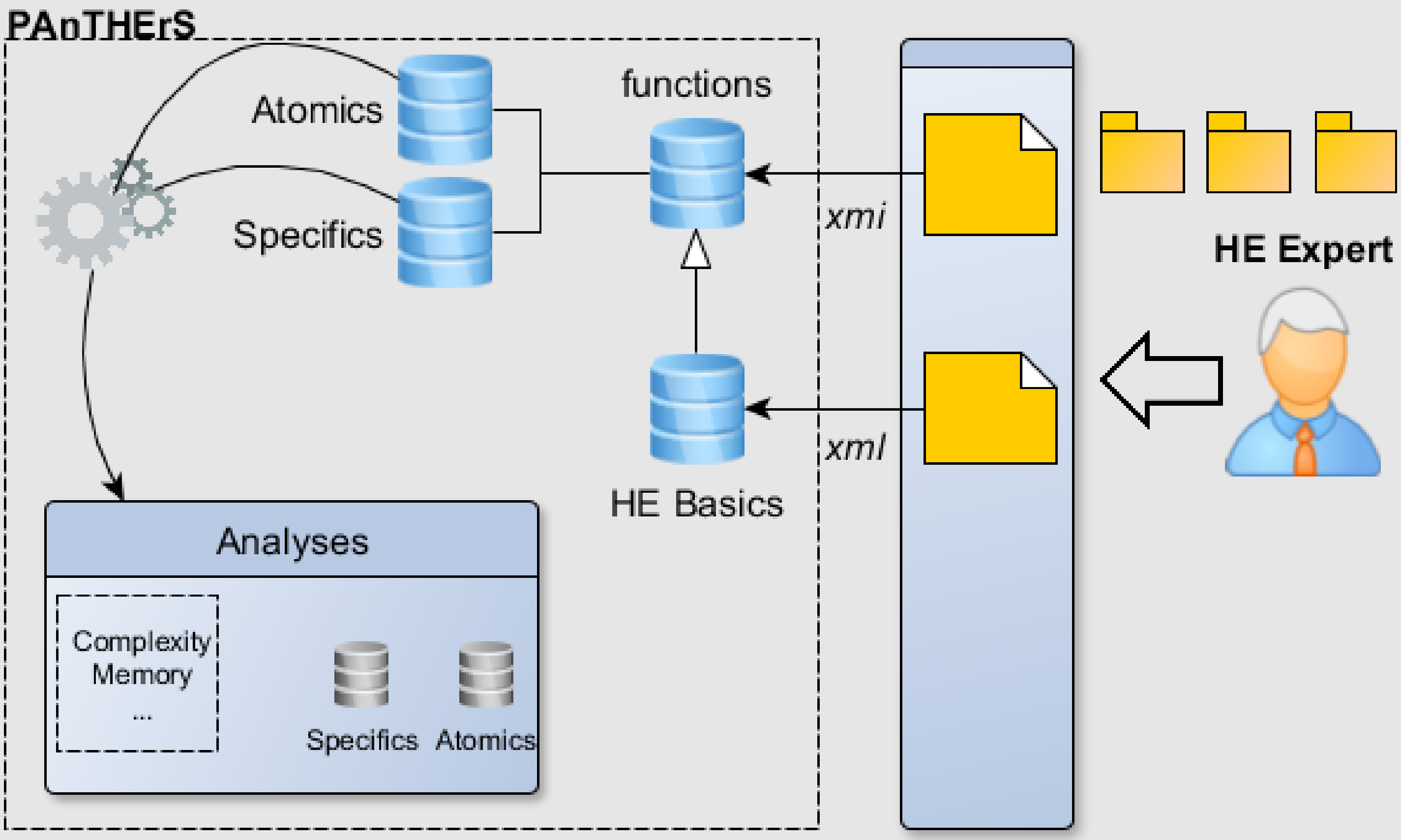
PAnTHErS usage



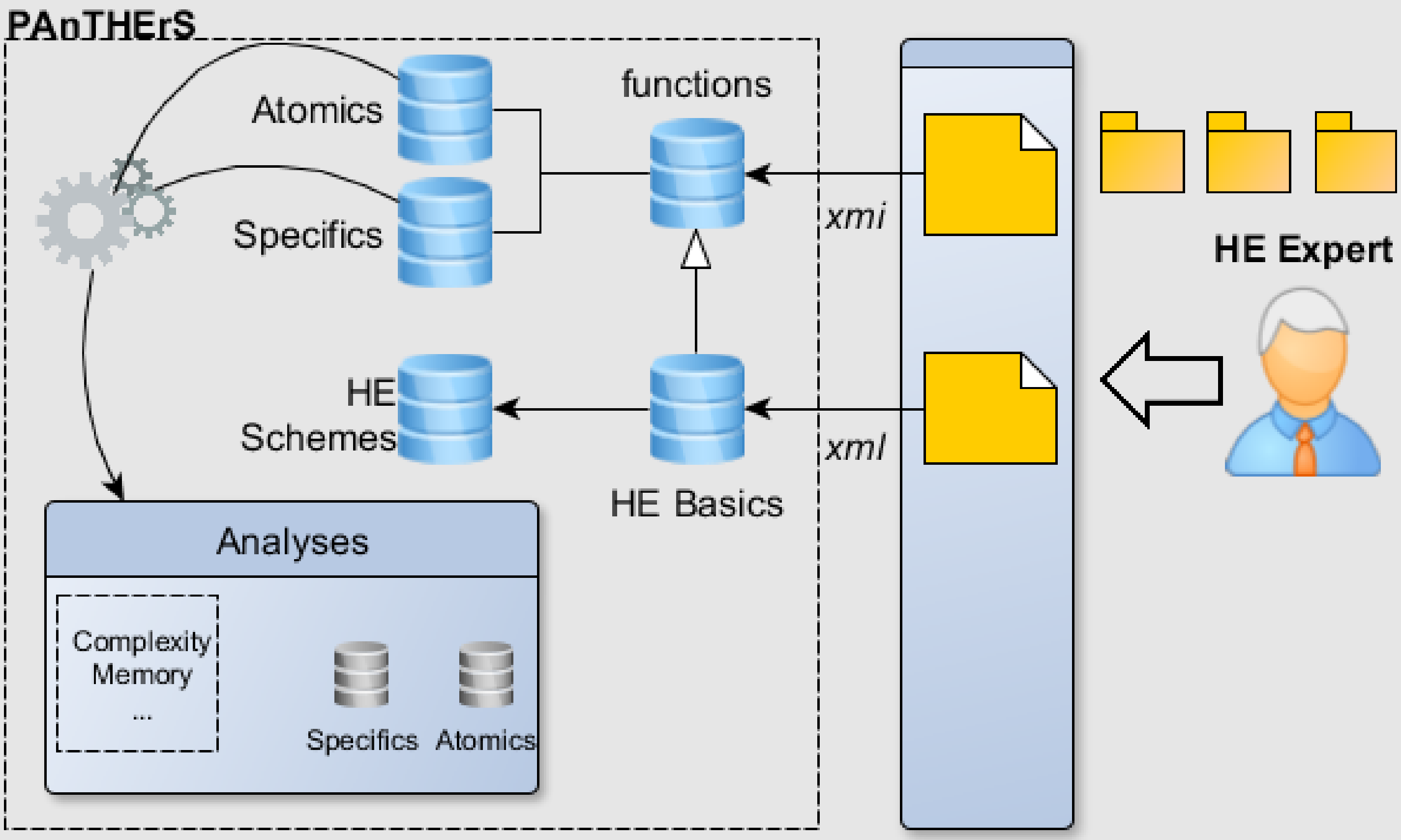
PAnTHErS usage



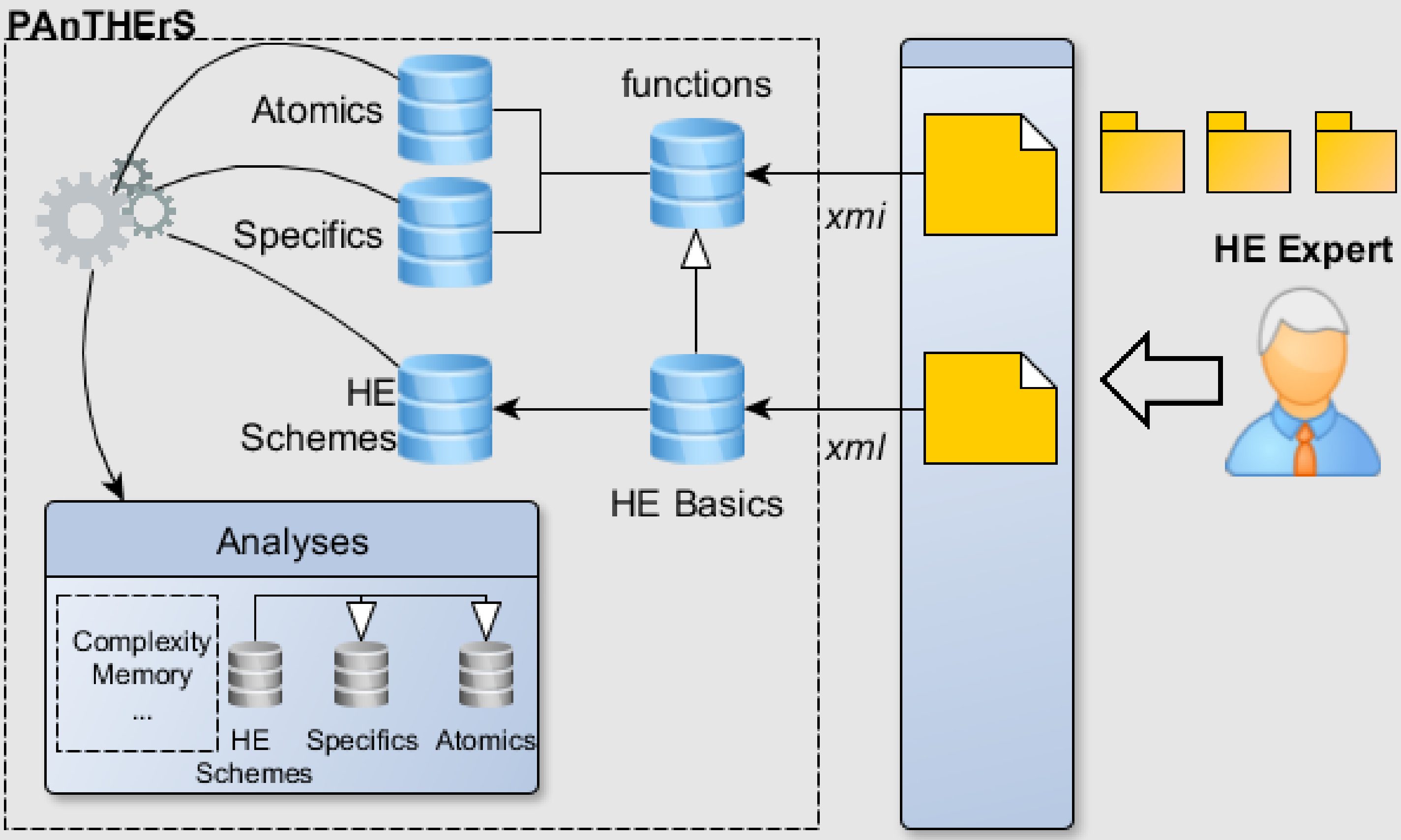
PAnTHErS usage



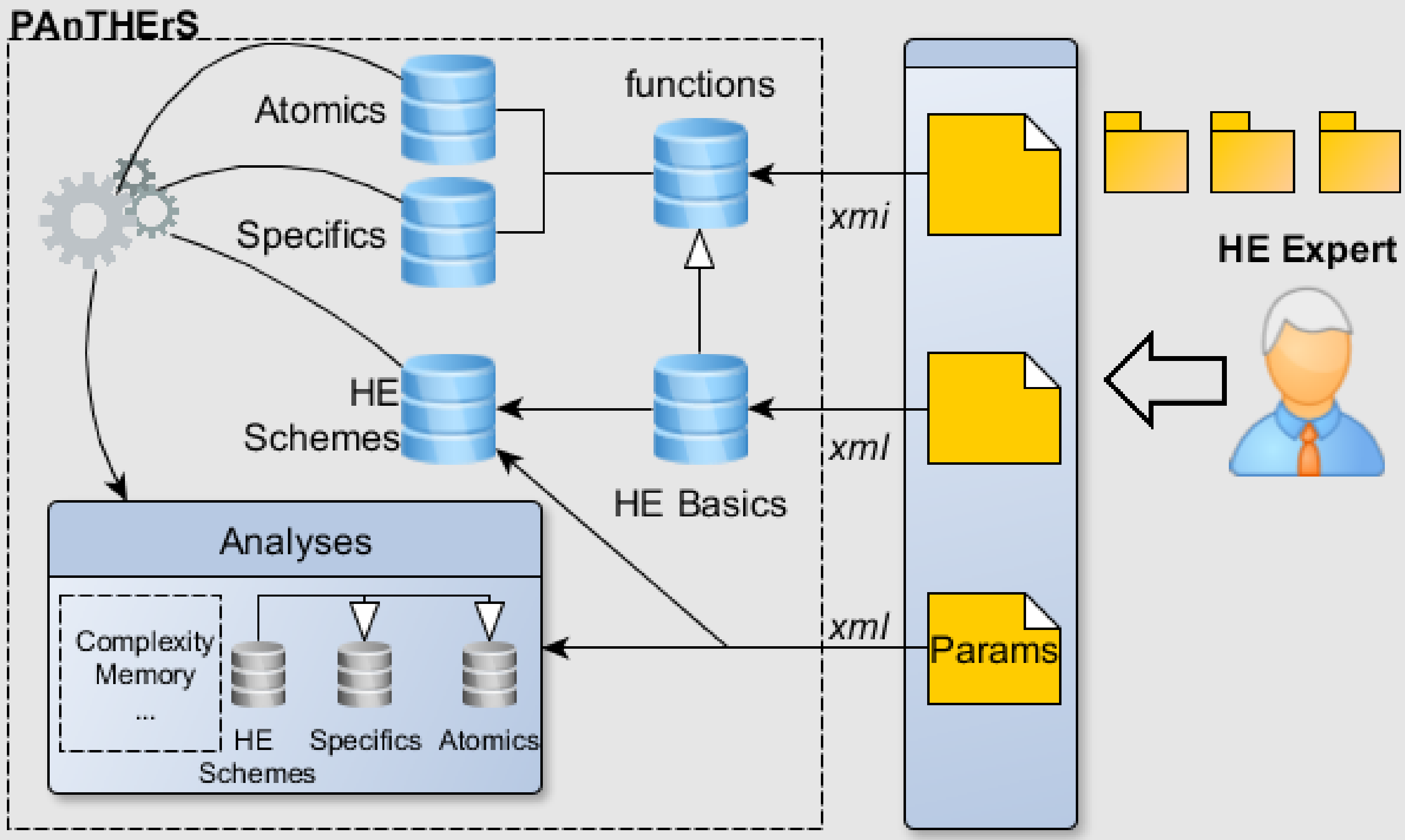
PAnTHErS usage



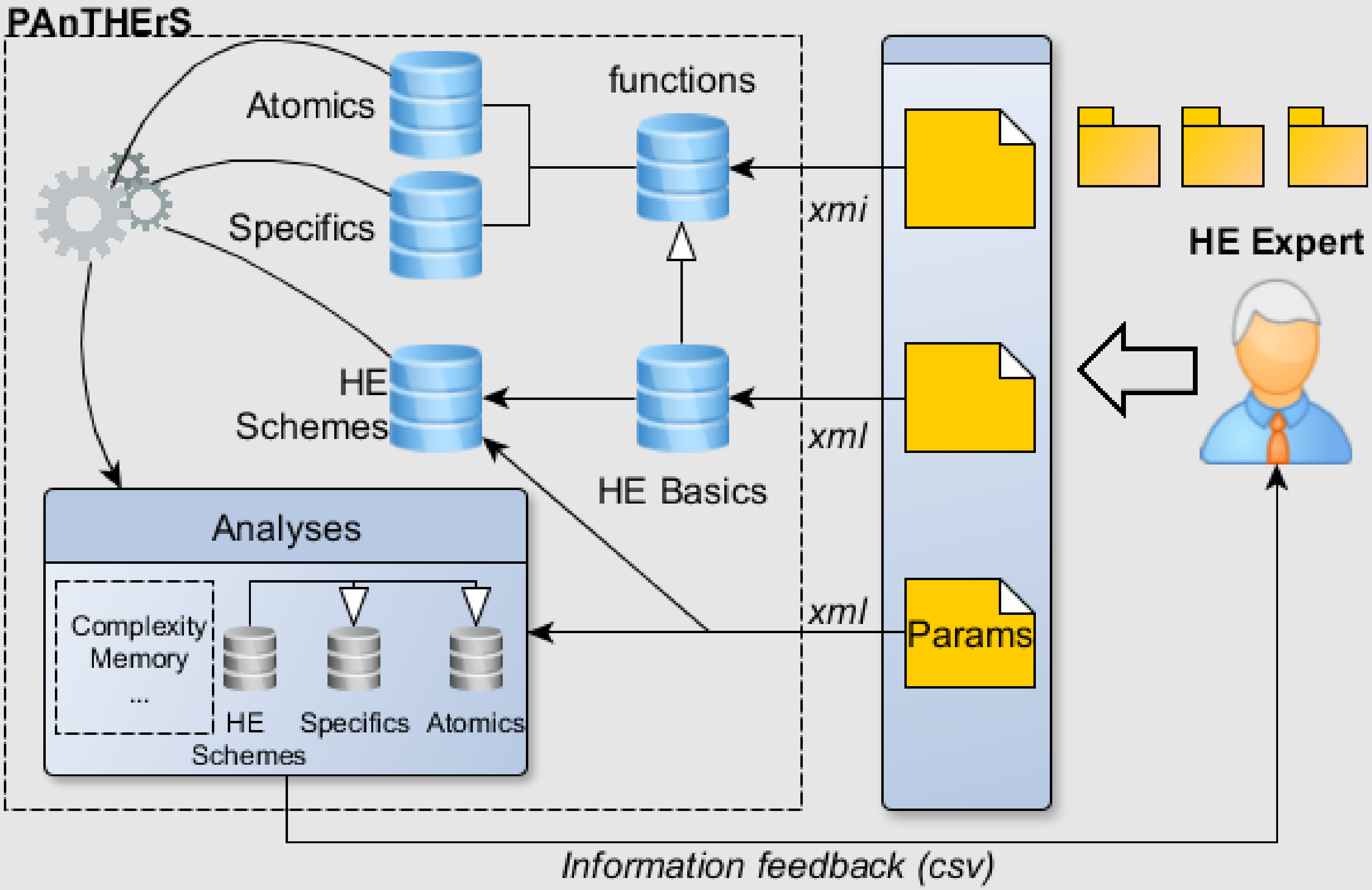
PAnTHErS usage



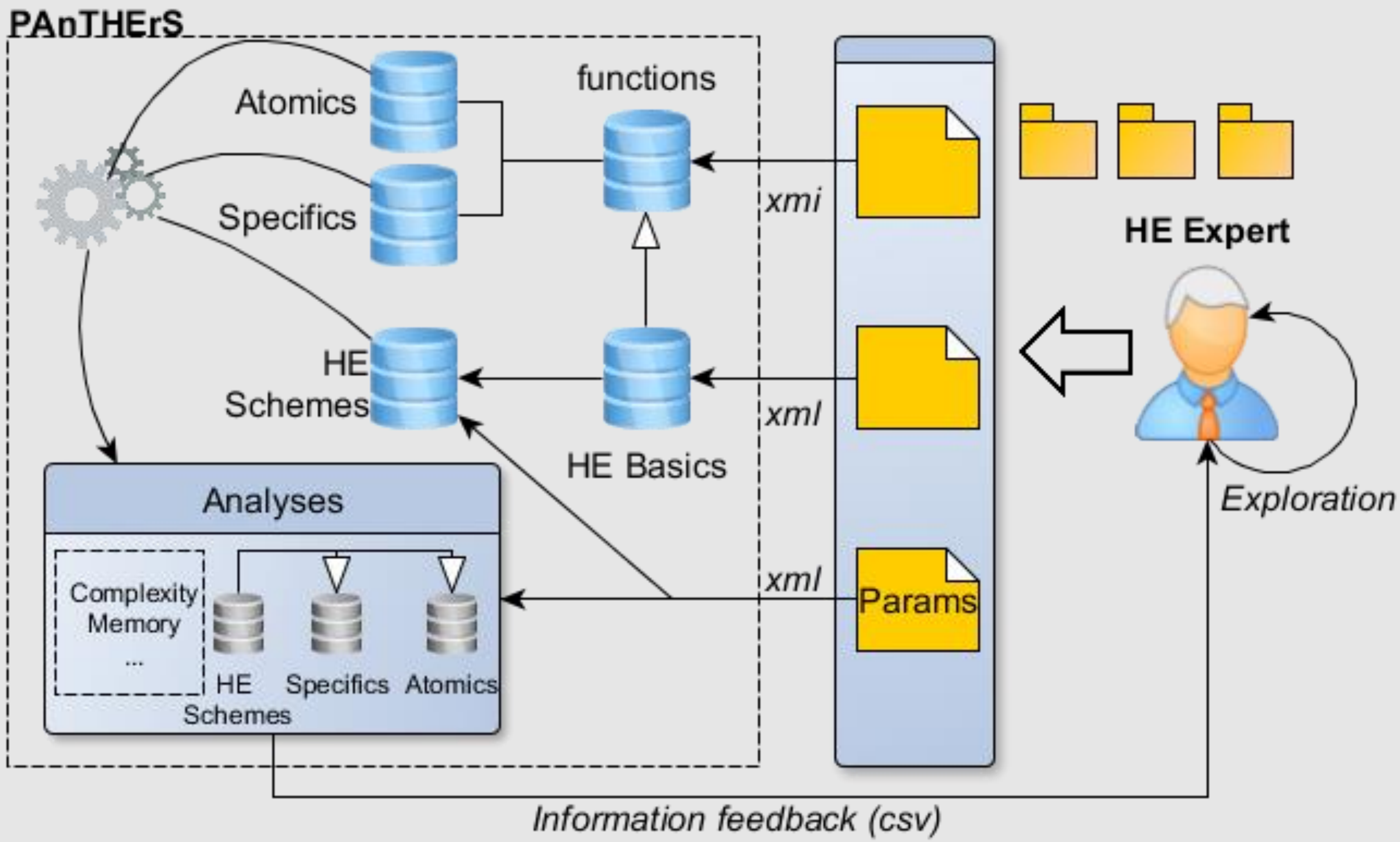
PAnTHErS usage



PAnTHErS usage



PAnTHErS usage



V – Results

1. FV and YASHE complexity
2. FV and YASHE memory cost
3. FV and YASHE depth

Results – *FV and YASHE complexity*

$R = \mathbb{Z}[x]/\Phi_d(x)$, $n = \varphi(d)$
$R_q = R/qR$
Integer base w
Plaintext space : $R_t = R/tR$

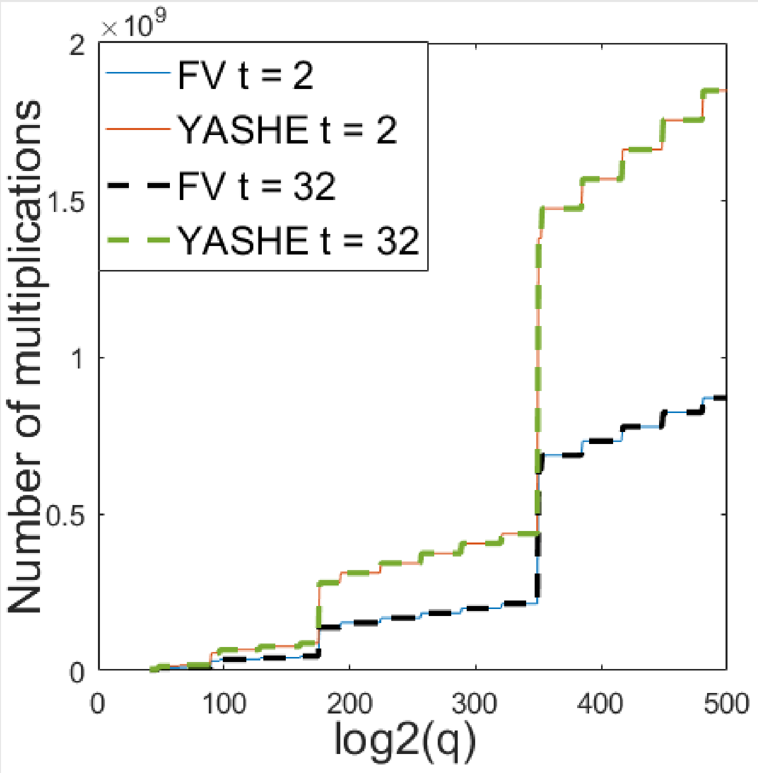
n	1024	2048	4096	8192
max. $\log_2(q)$	46	88	174	348

Ref : Migliore, V., Bonnoron, G., and Fontaine, C. (2017). Determination and exploration of practical parameters for the latest Somewhat Homomorphic Encryption (SHE) schemes. *Working paper or preprint.*

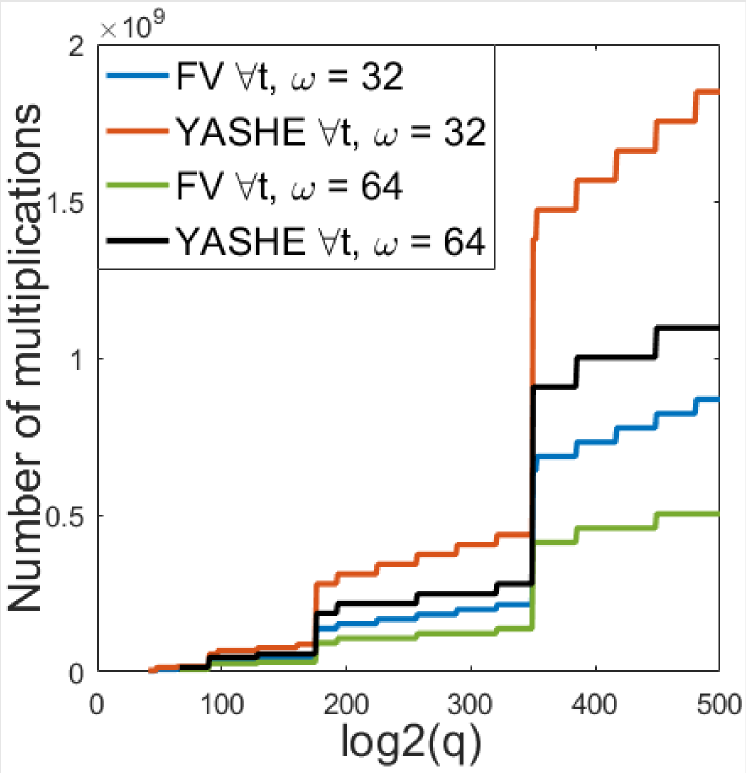
Results – *FV and YASHE complexity*

$R = \mathbb{Z}[x]/\Phi_d(x),\ n = \varphi(d)$
$R_q = R/qR$
Integer base w
Plaintext space : $R_t = R/tR$

n	1024	2048	4096	8192
max. $\log_2(q)$	46	88	174	348



Fixed parameter: $\log_2(w) = 32$



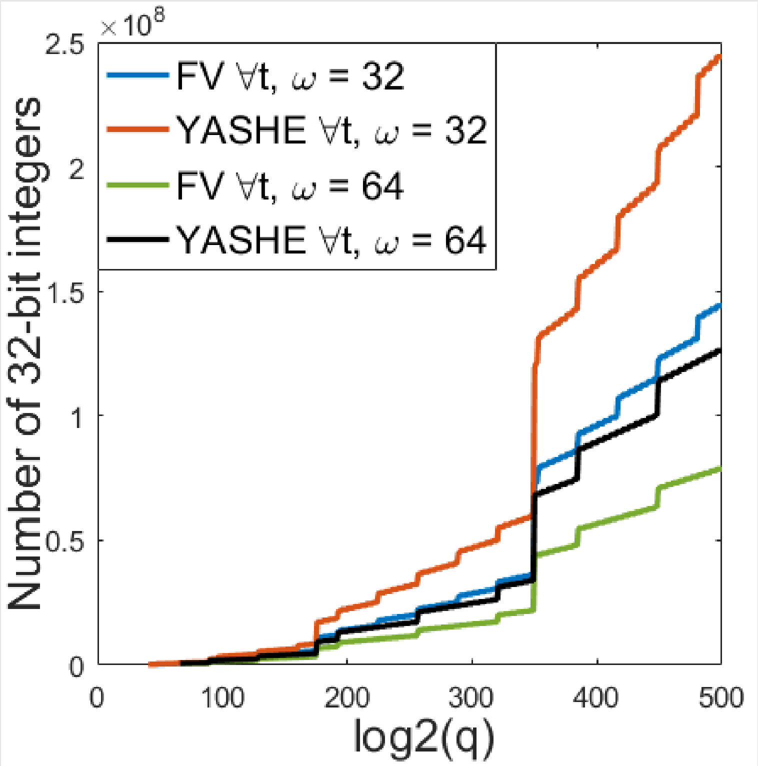
Here, $\omega = \log_2(w)$

Results – *FV and YASHE memory cost*

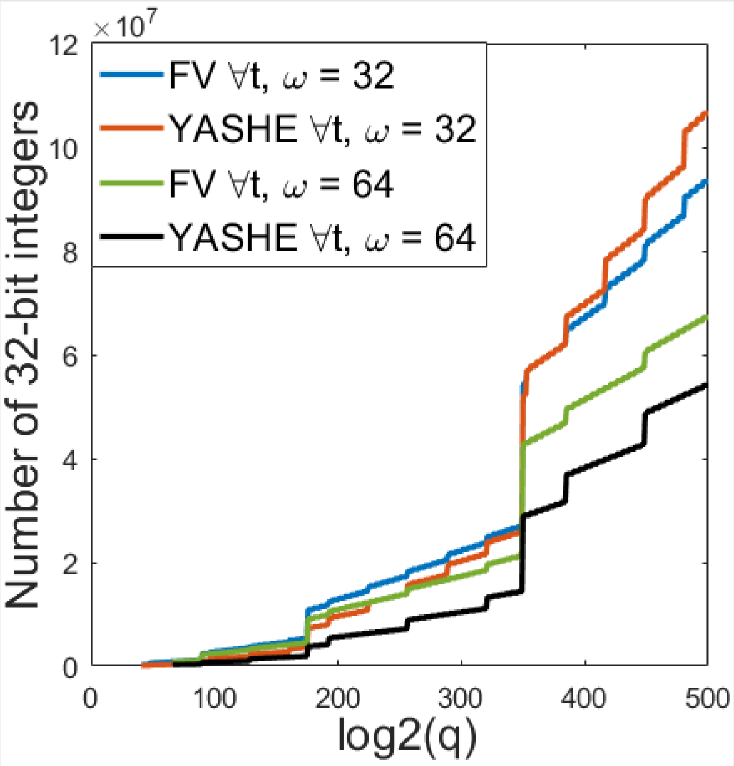
$R = \mathbb{Z}[x]/\Phi_d(x),\ n = \varphi(d)$
$R_q = R/qR$
Integer base w
Plaintext space : $R_t = R/tR$

n	1024	2048	4096	8192
max. $\log_2(q)$	46	88	174	348

Here, $\omega = \log_2(w)$



KeyGen function

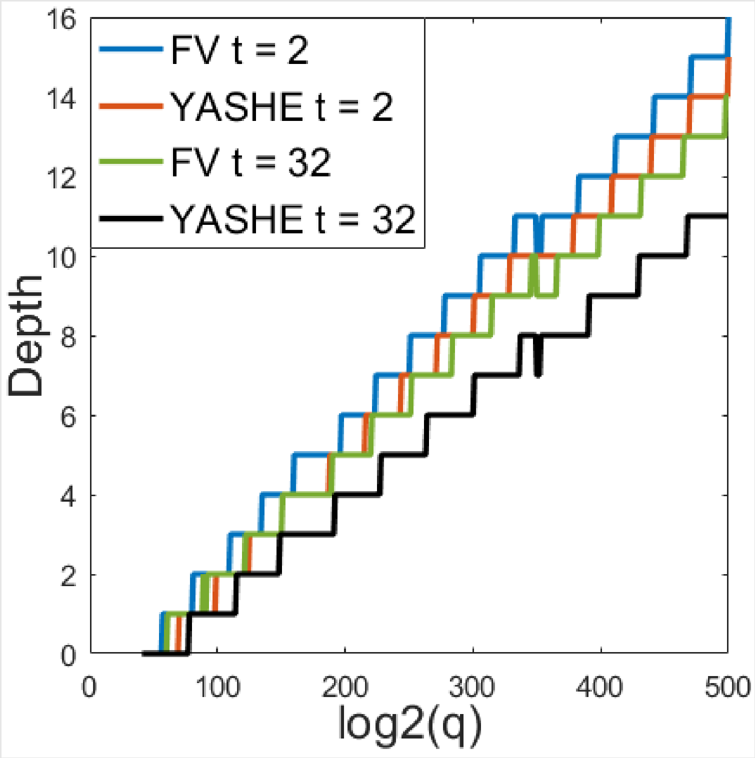


Mult function

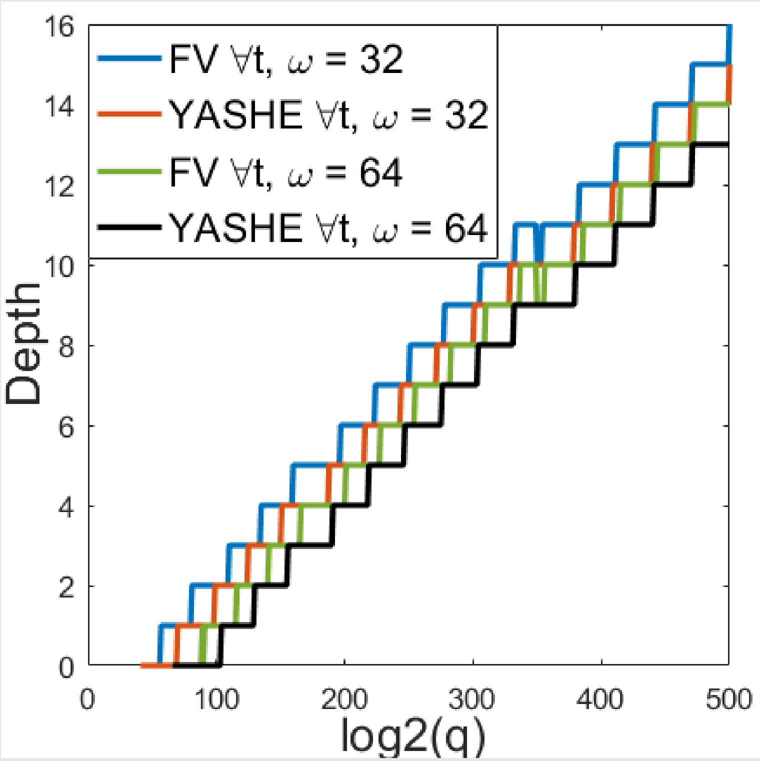
Results – *FV and YASHE depth*

$R = \mathbb{Z}[x]/\Phi_d(x),\ n = \varphi(d)$
$R_q = R/qR$
Integer base w
Plaintext space : $R_t = R/tR$

n	1024	2048	4096	8192
max. $\log_2(q)$	46	88	174	348



Fixed parameter: $\log_2(w) = 32$



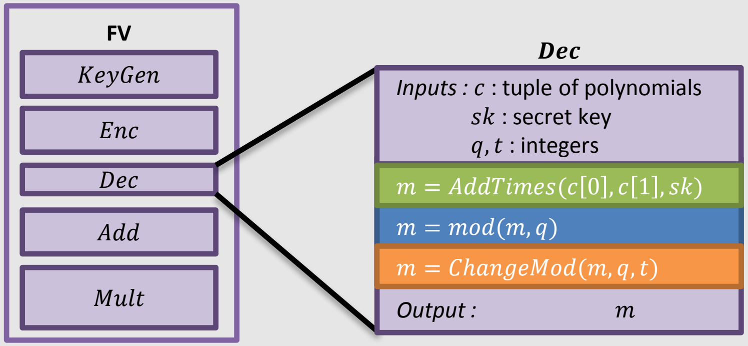
Here, $\omega = \log_2(w)$

VI – Conclusion & future work

Conclusion

HE scheme modeling :

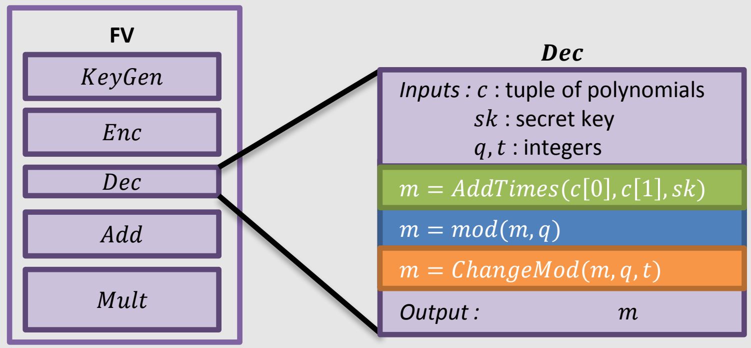
5 HE basic functions composed of Atomic and Specific functions.



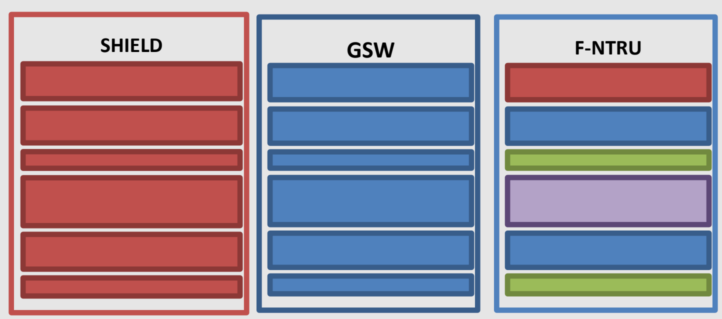
Conclusion

HE scheme modeling :

5 HE basic functions composed of Atomic and Specific functions.

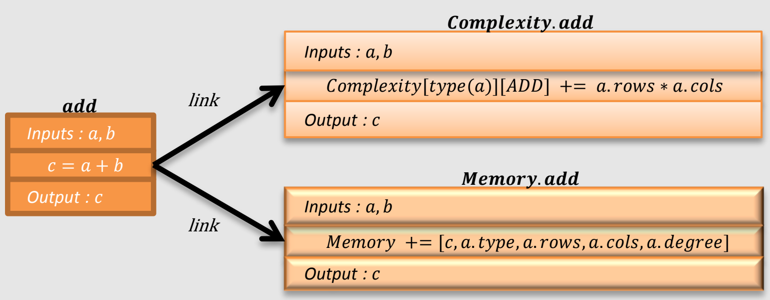


Filling the library implies **faster** modeling for new HE schemes.



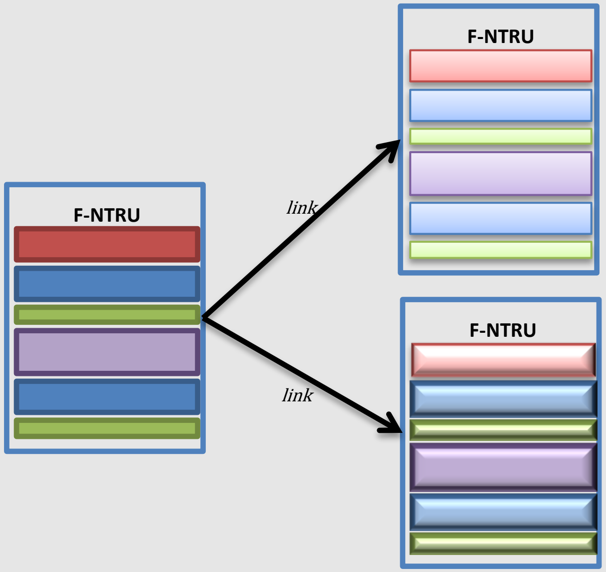
Conclusion

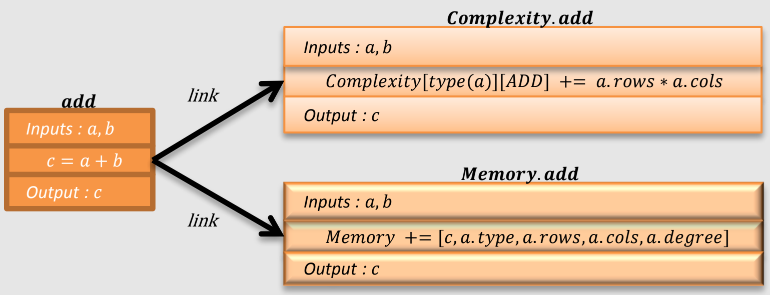
Atomic , specific & HE basic functions analysis:
Atomic, specific & HE basic functions are linked to their complexity and memory cost analysis functions.



Conclusion

Atomic , specific & HE basic functions analysis:
Atomic, specific & HE basic functions are linked to their complexity and memory cost analysis functions.

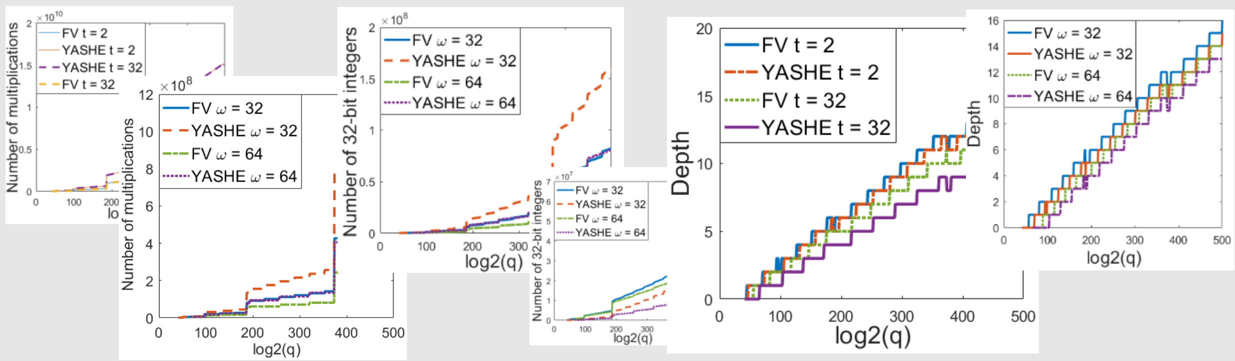




Automated generation of HE scheme **analysis functions.**

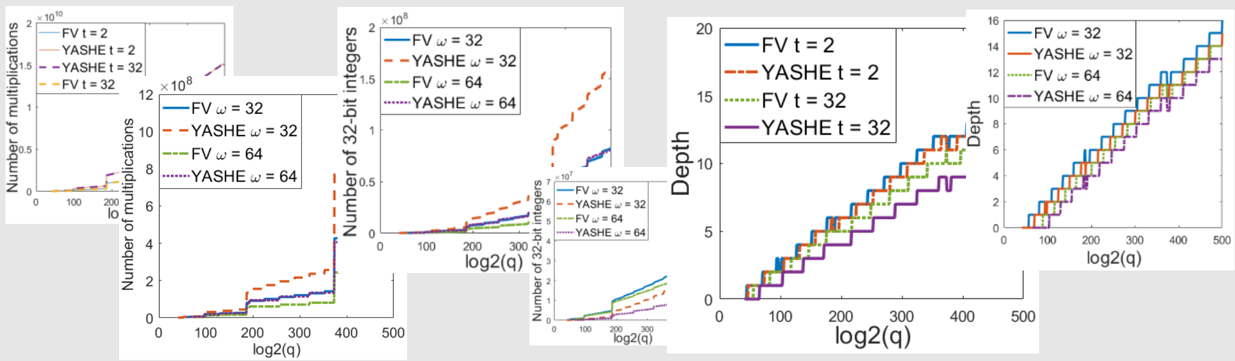
Conclusion

By **varying** input **parameters**, **PAnTHErS provides** memory, complexity and depth **results**.



Conclusion

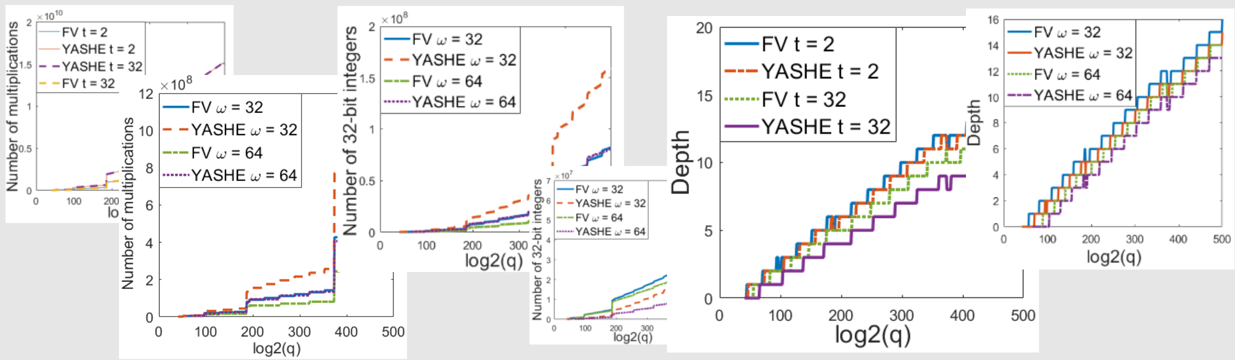
By **varying** input **parameters**, **PAnTHErS provides** memory, complexity and depth **results**.



HE expert interprets PAnTHErS data…

Conclusion

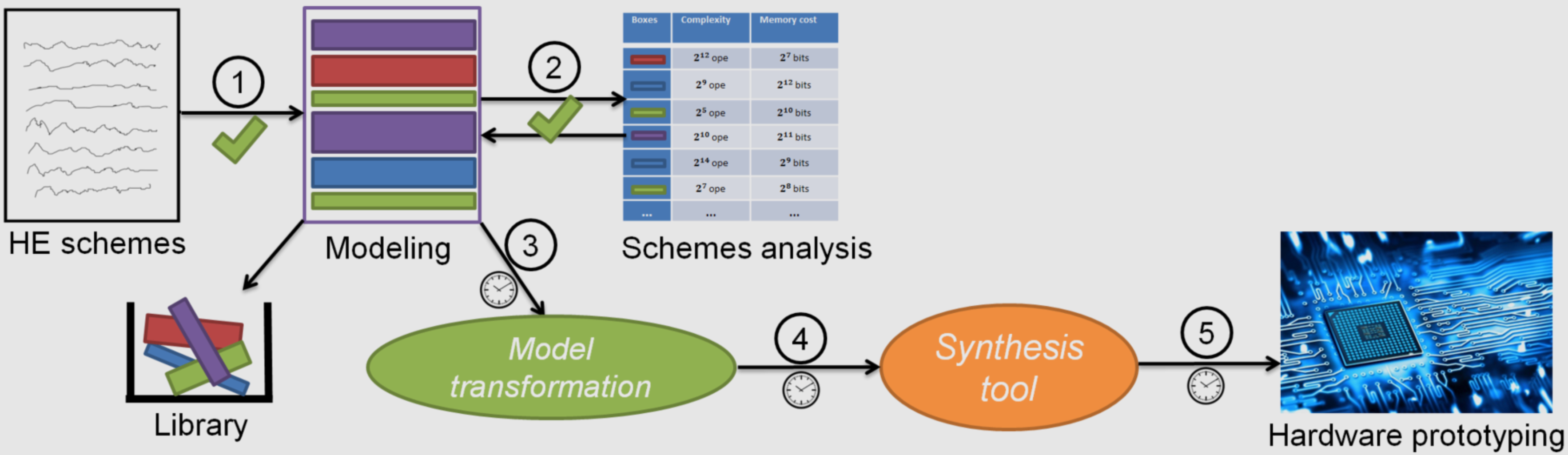
By **varying** input **parameters**, **PAnTHErS provides** memory, complexity and depth **results**.



HE expert interprets PAnTHErS data…

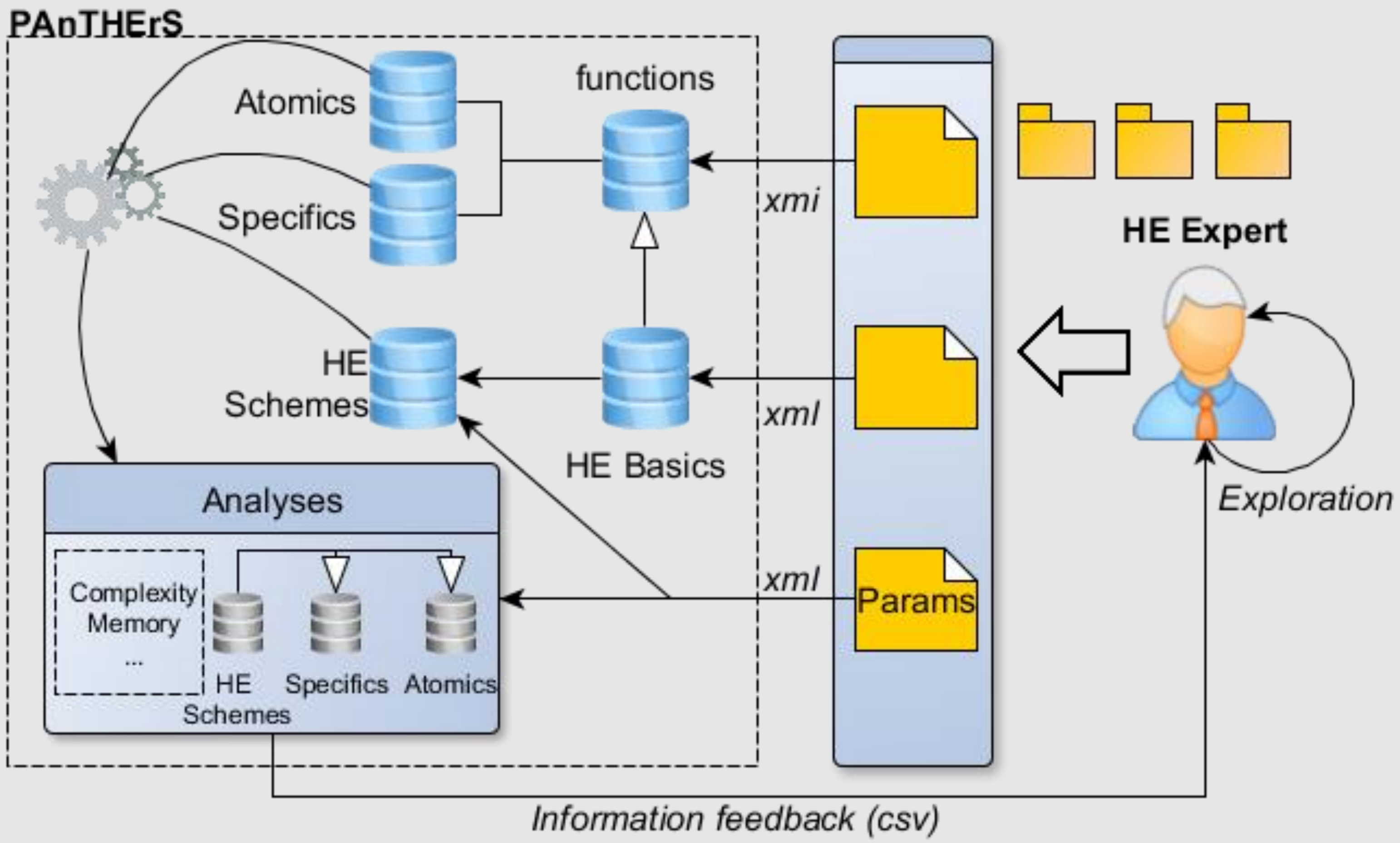
…to select the scheme which **fits better** his application.

Future work

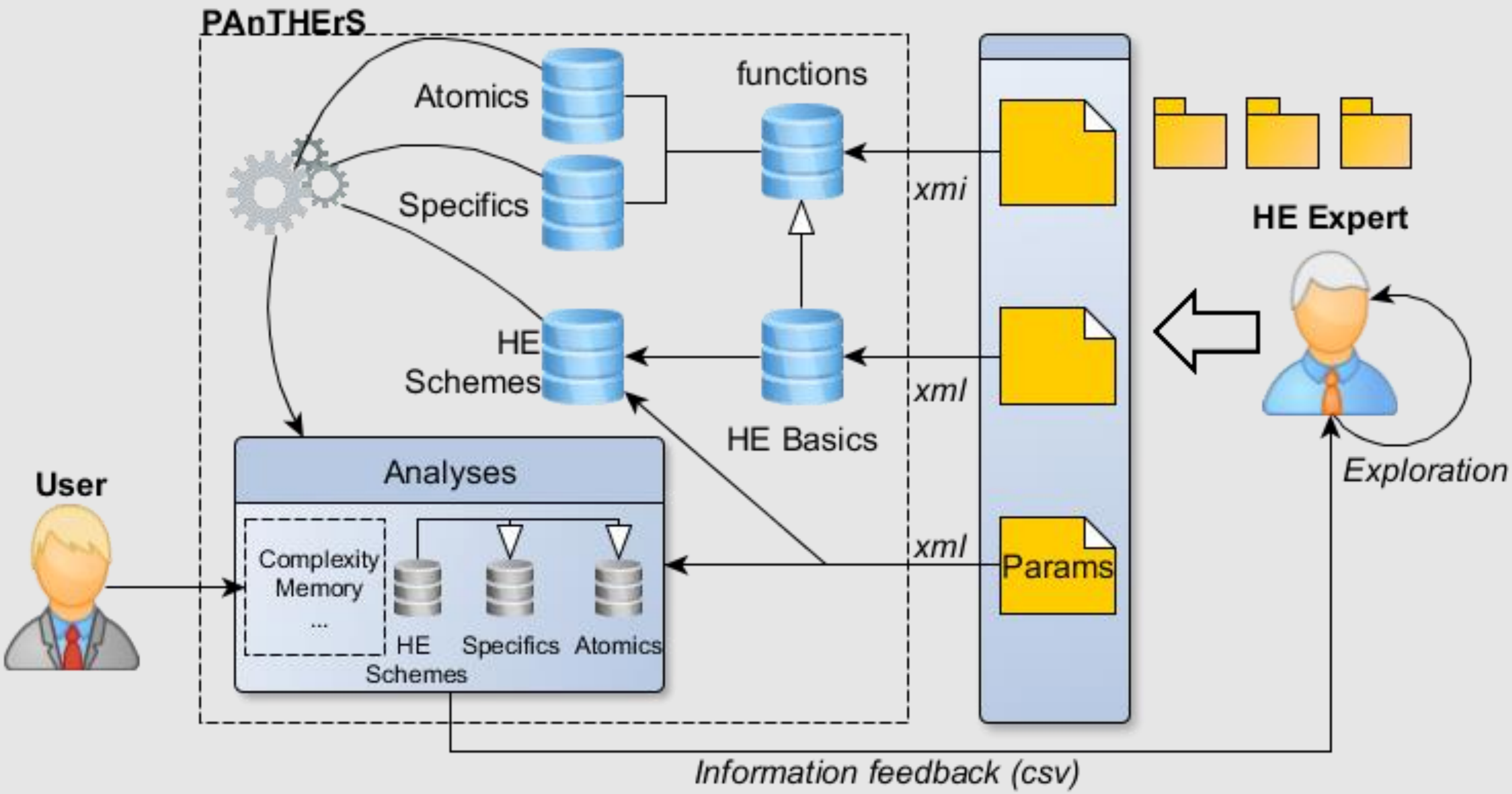


Boxes	Complexity	Memory cost
	2^{12} ope	2^{7} bits
	2^{9} ope	2^{12} bits
	2^{5} ope	2^{10} bits
	2^{10} ope	2^{11} bits
	2^{14} ope	2^{9} bits
	2^{7} ope	2^{8} bits
...	...	...

Future work


PAnTHErS
Atomics
Specifics
functions
HE
Schemes
HE Basics
xmi
xml
xml
Params
HE Expert
Exploration
Analyses
Complexity
Memory
...
HE
Schemes
Specifics
Atomics
Information feedback (csv)

Future work


PAnTHErS
Atomics
Specifics
functions
HE Schemes
HE Basics
xmi
xml
xml
Params
HE Expert
Exploration
User
Analyses
Complexity
Memory
...
HE Schemes
Specifics
Atomics
Information feedback (csv)

Thanks

Cyrielle FERON
Loïc LAGADEC
Vianney LAPOTRE

References :

[1] Migliore, V., Bonnoron, G., and Fontaine, C. (2017). Determination and exploration of practical parameters for the latest Somewhat Homomorphic Encryption (SHE) schemes. *Working paper or preprint.*
[2] Doröz, Y. and Sunar, B. (2016). Flattening NTRU for evaluation key free homomorphic encryption. *IACR Cryptology ePrint Archive*, 2016:315.
[3] Fan, J. and Vercauteren, F. (2012). Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptology ePrint Archive*, 2012:144.
[4] Gentry, C., Sahai, A., and Waters, B. (2013). Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In *Proc. CRYPTO*, pages 75–92, Santa Barbara, CA, USA.
[5] Khedr, A., Gulak, P. G., and Vaikuntanathan, V. (2016). SHIELD: scalable homomorphic implementation of encrypted data-classifiers. *IEEE Trans. Computers*, 65(9):2848–2858.
[6] Bos, J.W., Lauter, K. E., Loftus, J., and Naehrig, M. (2013). Improved Security for a Ring-Based FHE Scheme. In *Proc. Cryptography and Coding IMA*, pages 45–64, Oxford, UK.